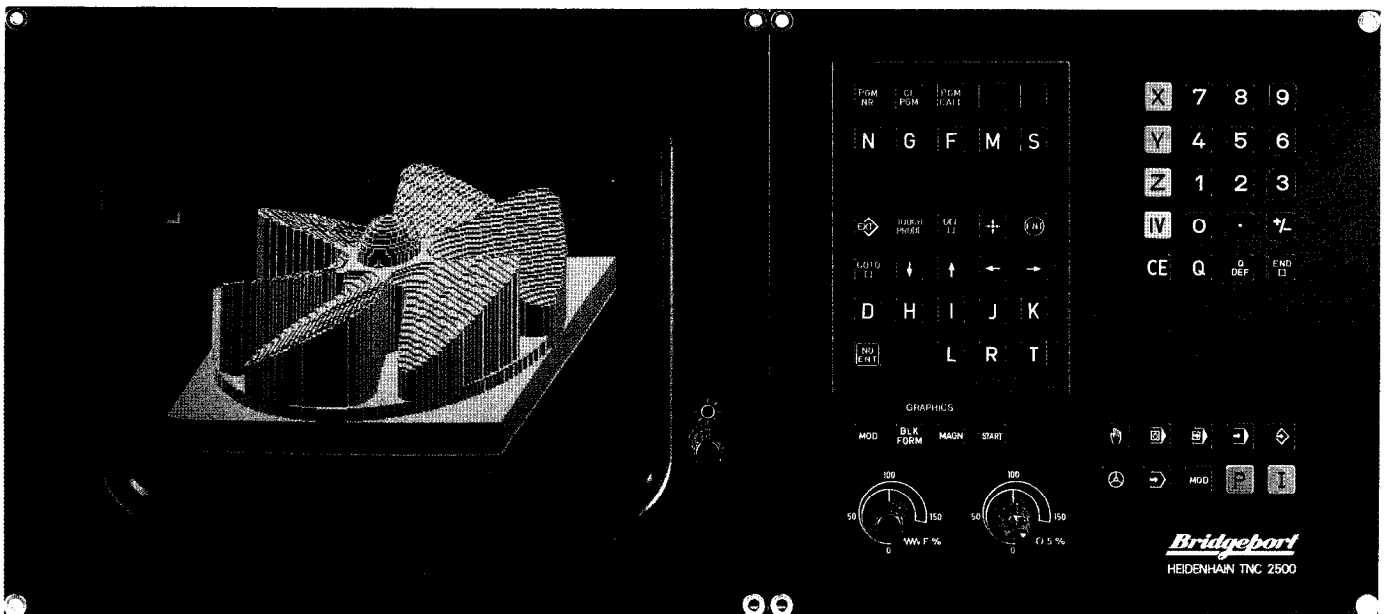


User Manual

ISO Programming

TNC 2500 B Contouring Control



Screen displays

PROGRAM RUN/FULL SEQUENCE									
Z7410 G71 *									
N10 G99 T1 L+0 R+2 *									
N20 T1 G17 S1000 *									
N25 G00 G40 G90 X+10 Y+10 M03 *									
N30 G54 X+100 Y+20 *									
N40 G28 X *									
N50 I+100 J+0 *									
N60 G73 G90 H+315 *									

ACTL. ☒ SN- 90,000 YN - 10,000									
Z + 1,560 R + 1,000									
CC X + 0,000 ROT + 45,000									
Y + 20,000 SCL 0,800000									
T1 Z S 1000 F M3/9									

Operating mode
Error messages/dialog line

Preceding block
Current block
Next block
Block after next

Status display

Status display:

ACTL: Type of position display, switchable with MOD
(further displays: NOML, DIST., LAG – see index “General Information”)

X ...
Y ...
Z ...
etc. } Position coordinates

*: “Control in operation” display

↔: “Axis is locked” display

N: Datum shift, shown as an index on the shifted axis.

S: Mirror image, shown as an index on the mirrored axis.

ROT: Basic rotation of the coordinate system

SCL: Scaling

CC: Circle center or pole

T ...: Called tool

Z: Spindle axis

S: Spindle speed

F: Feed rate

M: Spindle status (M03, M04, M05, M13, M14)

Guideline for procedure from preliminary operations to workpiece machining

Sequence	Action	Operating mode	Cross reference	Page
1	Select tools	—	Workpiece drawing	—
2	Set datum for workpiece machining	—	Workpiece coordinates	A15
3	Determine speeds and feed rates	—	Spindle speed, feed rate diagrams	A20
4	Switch on machine	—	Machine operating manual	—
5	Traverse reference points (homing the machine)	—	Switch on	M1
6	Clamp workpiece	—	Clamping instructions	—
7a	With 3D Touch Probe: datum setting and compensation of workpiece misalignment	 Manual	Workpiece setup with the 3D Touch Probe	M3
or				
7b	Align workpiece, insert zero tool, mark workpiece and set datum	 Manual	Manual operation Machine handbook: Tool change	M13
8	Enter program – via keyboard or from external storage	 Programming and editing	Back fold-out page, program example; Programming and editing	P1
9	Test program (without axis movements)	 Test run	Programming, Test run	P124
10	Graphic program simulation (without axis movements)	 Program run	Programming, Graphic simulation	P125
11	Test run without tool in single block mode	 Program run, Single block	Program run	M19
12	Optimize program if necessary	 Programming and editing	Programming and editing	P3
13	Insert tool and machine workpiece automatic program run	 Program run, Full sequence	Program run	M19

Operating Panel TNC 2500 B

with snap-on keypad

Machine Operating Modes

-  Manual operation
-  Electronic handwheel
-  Positioning with manual data input
-  Program run, Single block
-  Program run, Full sequence

Programming Modes

-  Programming and editing
-  Test run with graphic simulation

Program Management

-  Naming/selecting a program
-  Clear program
-  Programmable program call
-  External program input and output
-  Supplementary operating modes

Graphics

-  Graphic operating modes
-  Define blank form, reset blank form
-  Magnify detail
-  Start graphic simulation

Override

-  $\circ S\%$ Feed rate override
-  $\circ\circ\circ F\%$ Spindle speed override

Screen control

-  brightness

Programming in ISO Format

-  Block number
-  G code
-  Feed rate/Dwell time with G04/Scaling factor
-  Miscellaneous function
-  Spindle speed in rpm
-  Parameter definition
-  Polar coordinate angle/
Angle of rotation in cycle G73
-    X, Y, Z coordinates of a circle center
-  Set label number with G98/
Jump to label number/
Tool length with G99
-  Polar coordinate radius/
Rounding-off radius with G25, G26, G27/
Chamfer with G24/
Circle radius with G02, G03, G05
Tool radius with G99/
-  Tool definition with G99/
Tool call

Entering and Editing Values

-  ...  Axis keys
-  ...  Number keys
-   Decimal point, sign change
-  Key for polar coordinates
-  Key for incremental dimensions
-   Enter parameter instead of a number,
Define parameter
-  Transfer actual position to memory
-    Cursor keys,
Jump to a certain block or cycle
-    No entry, Enter data,
Terminate block entry
-  Clear entry
-  Delete block

Contents

General Information	Introduction	A1
	MOD Functions	A8
	Coordinates	A15
	Linear and Angle Encoders	A18
	Cutting Data	A20
Machine Operating Modes	Switch-On	M1
	Manual Operation	M2
	3D Touch Probe	M3
	Datum Setting	M13
	Electronic Handwheel, Incremental Jog	M15
	Positioning with Manual Data Input	M17
	Program Run	M19
Programming Modes	Programming in ISO	P1
	Program Selection	P6
	Tool Definition	P10
	Cutter Path Compensation	P15
	Tools	P18
	Feed Rate F/Spindle Speed S/Miscellaneous	
	Functions M	P20
	Programmable Stop/Dwell Time	P21
	Path Movements	P22
	Linear Movement, Cartesian	P25
	Circular Movement, Cartesian	P30
	Polar Coordinates	P41
	Contour Approach and Departure	P48
	Predetermined M Functions	P51
	Program Jumps	P55
	Program Calls	P64
	Standard Cycles	P65
	Coordinate Transformations	P93
	Other Cycles	P102
	Parametric Programming	P105
	Programmed Probing	P120
	Teach-In	P123
	Test Run	P125
	Graphic Simulation	P126
	External Data Transfer	P129
	Address Letters in ISO	P137

General Information (A)

Introduction	1
Brief description of TNC 2500B	3
Machine operating modes	4
Programming modes	5
Accessories: 3D Touch Probe Systems	6
FE 401 Floppy Disk Unit	7
HR 130/HR 330 Electronic Handwheels	7
MOD Functions	8
Position displays	9
Traverse range limits	10
User parameters	11
Coordinates	
The coordinate system	15
Datum	16
Absolute and incremental coordinates	17
Linear and Angle Encoders	18
Cutting Data	
Feed rate diagram	20
Spindle speed diagram	21
Feed rate diagram for tapping	22

Introduction

Description

The TNC 2500B from HEIDENHAIN is a shop-floor programmable contouring control with up to 4 axes for milling and boring machines as well as for machining centers. It is conceived for the "man at the machine", featuring conversational programming and excellent graphic simulation of workpiece machining. Its background programming feature permits a new program to be created (or a program located in memory to be edited) while another program is being executed. Besides fixed cycles, coordinate transformations and parametric programming, the control also includes functions for 3D touch probes.

Programs can be output to peripheral devices and read into the control via the RS-232-C data interface, allowing programs to be created and stored externally.

Conversational or ISO programming

In addition to programs written in conversational format, ISO programs can also be entered, either via the snap-on keyboard or via the data interface. Both interactive format and ISO format programs can reside in memory at the same time.

Compatibility

This control can execute programs from other HEIDENHAIN controls, provided they contain only the functions described in this manual.

Structure of manual

This manual addresses the skilled machine operator and requires appropriate knowledge of non-NC-controlled boring and milling.

TNC beginners are advised to work through this manual and the examples systematically. If you have already worked with a HEIDENHAIN TNC, you can skip familiar topics.

This manual deals with **programming in ISO format**. HEIDENHAIN conversational programming is described in detail in a separate user manual for the TNC 2500B.

The sequence of chapters in this operating manual is according to control operating modes and key functions, as well as according to the logical working order:

- **Machine operating modes:**
Switch-on – setup – set display value – machine workpiece.
- **Programming modes:**
Programming and editing – test run.

Symbols for keys

The following symbols are used in this manual:

Empty square:		... keys for numerical input on the TNC operating panel
Square with symbol, e.g.		... other keys on the TNC operating panel
Circle with symbol, e.g.		... buttons on the machine operating panel

The pages of this manual are distinctly marked with the relevant key symbols.

Typeface for screen displays

Program blocks and TNC screen dialogs are printed in this **SPECIAL TYPE**.

Introduction

Program Examples

The example programs in this manual are based on a uniform blank size and can be displayed on the screen by adding the following blank definition (see index "Programming Modes", Program Selection):

```
G30 G17 X+0 Y+0 Z-40
G31 G90 X+100 Y+100 Z+0
```

The examples can be executed on machine tools with tool axis Z and machining plane XY. If your machine uses a different axis as the tool axis, this axis must be programmed instead of Z and likewise the corresponding axes for the machining plane.



Beware of collisions when executing the example programs!

Buffer batteries in the control

Buffer batteries protect the stored programs and machine parameters against loss due to power interruption.

When the message
EXCHANGE BUFFER BATTERY
appears, you must change the batteries.

The batteries should be replaced once a year.

Battery type:
3 AA-size batteries, leak-proof
IEC designation "LR6"

Changing the battery

Battery replacement is described in the manual of the machine manufacturer.

Error messages

The TNC checks input data and status of the control and machine.

	Cause and reaction of the control:	Remedy:
Input range exceeded	The permitted range of values is exceeded: e.g. feed rate too high. The value is not accepted and an error message appears.	Clear the value with the "CE" key, enter and confirm the correct value.
Incompatible/contradictory inputs	E.g. G00 X+50 X+100 During "TEST RUN" or during program execution, the TNC stops with an error message before executing the corresponding block and displays the block number in which an error was found.	Change to the "PROGRAMMING AND EDITING" operating mode. The error can normally be found either in the block with the displayed block number or in a previously executed block. Then: correct the error. Operating mode "Full sequence" and restart.
Malfunction of the machine or control	Malfunctions that affect operating safety cause blinking error messages. Note down the error message!	Switch off the machine or the control. Remove the fault if possible. Attempt to restart. If the program then runs correctly, the problem was only a spurious malfunction. If the same error message comes up again, contact the customer service of the machine manufacturer.

TNC 2500 B

Brief description

Control type	Contouring control for 4 axes
Traversing possibilities	Straight lines in 3 axes Circles in 2 axes Helix
Background programming	Programming and program execution simultaneously
Graphics	Graphic simulation in the "Program run" operating modes
Program input	In HEIDENHAIN format or according to ISO
Input resolution	Max. 0.001 mm or 0.0001 inch or 0.001°
Program memory	For 32 programs, battery buffered: 4000 program blocks
Tools	Up to 254 tool definitions in a program Up to 99 tools in the central tool file

Programmable functions

Contour	Straight line, chamfer Circle (input: center and end point of the arc or radius and end point of the arc), circle connected tangentially to the contour (input: arc end point) Corner rounding (input: radius) Tangential approach and departure from a contour
Program jumps	Subprograms, program section repeats, call of other programs
Fixed cycles	Drilling cycles for pecking, tapping Milling cycles for rectangular pocket, circular pocket, slot "Subcontour List" cycles for milling pockets and islands with irregular contours
Coordinate transformations	Move and rotate the coordinate system, mirror image, scaling
Probing functions	For 3-D touch trigger probe
Parameter programming	Mathematical functions (= / + / - / x / ÷ / sin / cos / angle α from axis sections / $\sqrt{a}$ / $\sqrt{a^2 + b^2}$); parameter comparison (= / ≠ / > / <)
Traversing range	Max. $\pm 30\,000$ mm or 1180 inches
Cutting data	Traversing speed: max. 30 m/min or 1180 inches/min Spindle speed: max. 99 999 rpm

Hardware

Component units	Logic unit, control panel and monochrome screen
Block processing time	1500 blocks/min (40 ms)
Control loop cycle time	6 ms
Data interface	RS-232-C/V.24 Data transfer speed: max. 19 200 baud
Ambient temperature	Operation: 0° C to 45° C (32° F to 113° F) Storage: -30° C to 70° C (-22° F to 158° F)

Machine operating modes



Manual operation



The axes can be moved via the external axis direction buttons. Workpiece datum can be set as desired.

MANUAL OPERATION			
DATUM SET		Z =	
ACTL.	X	+	49,258
	Y	+	23,254
	Z	+	15,321
		F 0	M5/9

Electronic Handwheel



The axes can be moved either via an electronic handwheel or via the external axis direction buttons. It is also possible to position by defined jog increments.

HANDWHEEL ?			
INTERPOLATION FACTOR:		5	6
ACTL.	X	+	49,258
	Y	+	23,254
	Z	+	15,321
		F 0	M5/9

Positioning with manual data input (MDI)



The axes are positioned according to the data keyed in. These data are not stored.

POSITIONING MANUAL DATA INPUT			
N10 G07 X+20 F200 M03			
ACTL.	X	+	9,375
	Y	+	8,200
	Z	+	8,985
	A	+	0,180
T	F		M5/9

Program run

A part program in the memory of the control is executed by the machine.

Full sequence



After starting via the machine START button, the program is automatically executed until the end or a STOP is reached.

Single block



Each block is started separately with the machine START button.

PROGRAM RUN/FULL SEQUENCE			
Z7410 G71 *			
N10 G99 T1 L+0 R+2 *			
N20 T1 G17 S1000 *			
N25 G00 G40 G90 X+10 Y+10 M03 *			
N30 G54 X+100 Y+20 *			
N40 G28 X *			
N50 I+100 J+0 *			
N60 G73 G90 H+315 *			

ACTL.	X +	9,375	Y + 8,200
	Z +	8,985	R + 0,180
T F 0 M5/9			

Programming modes



Programming and editing



Part programs can be entered, looked over and altered in the "Programming and editing" operating mode.

In addition, programs can be read in and output via the RS-232-C data interface.

```
PROGRAMMING AND EDITING
TOOL RADIUS R ?
Z7410 G71 *
N10 G99 T1 L+0 R+2 *
N20 T1 G17 S1000 *
N25 G00 G40 G90 X+10 Y+10 M03 *
N30 G54 X+100 Y+20 *
N40 G28 X *
N50 I+100 J+0 *
N60 G73 G90 H+315 *
-----
ACTL. X + 9,375 Y + 8,200
      Z + 8,985 A + 0,180

T                                F 0      MS/9
```

Test run



In the "Test run" operating mode, machining programs are analyzed for logical programming errors, e.g. exceeding the traversing range of the machine, redundant programming of axes, certain geometrical incompatibilities etc.

```
TEST RUN
TO BLOCK NUMBER =
Z7410 G71 *
N10 G99 T1 L+0 R+2 *
N20 T1 G17 S1000 *
N25 G00 G40 G90 X+10 Y+10 M03 *
N30 G54 X+100 Y+20 *
N40 G28 X *
N50 I+100 J+0 *
N60 G73 G90 H+315 *
-----
ACTL. X + 9,375 Y + 8,200
      Z + 8,985 A + 0,180

T                                F 0      MS/9
```

Test graphics



GRAPHICS



In the "Program run" operating modes "full sequence" and "single block", you can graphically simulate machining programs via the "GRAPHICS" keys.

Display modes:

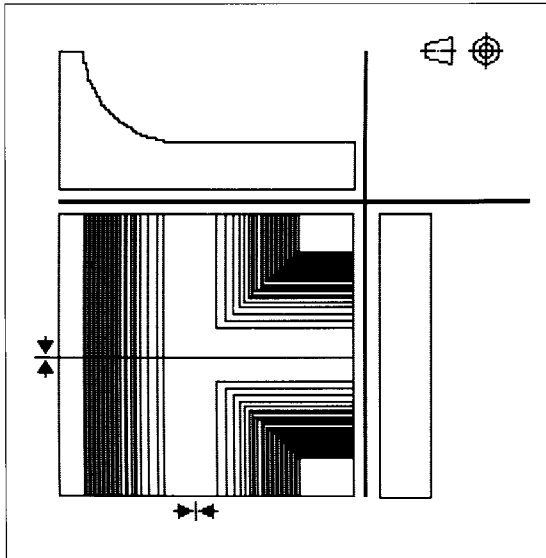
- plan view with depth indication
- view in three planes
- 3-D view

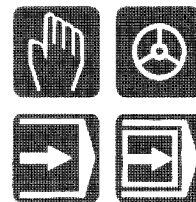
External data transfer



In the "Programming and editing" mode, programs can be read-in from an external storage medium and read-out to an external unit. Data transfer takes place via the RS-232-C data interface.

In the "Program run, single block" and "Program run, full sequence" modes of operation it is possible to read-in programs whose size exceeds the control's memory block by block for simultaneous execution.





The TNC software incorporates measuring cycles for the application of a HEIDENHAIN 3D Touch Probe in the "Manual", "Handwheel" and "Program run" operating modes.

Manual use



The following measurements can be performed in the "Manual" and "Handwheel" operating modes:

- position
- line
- angle
- corner point
- circle radius and circle center.

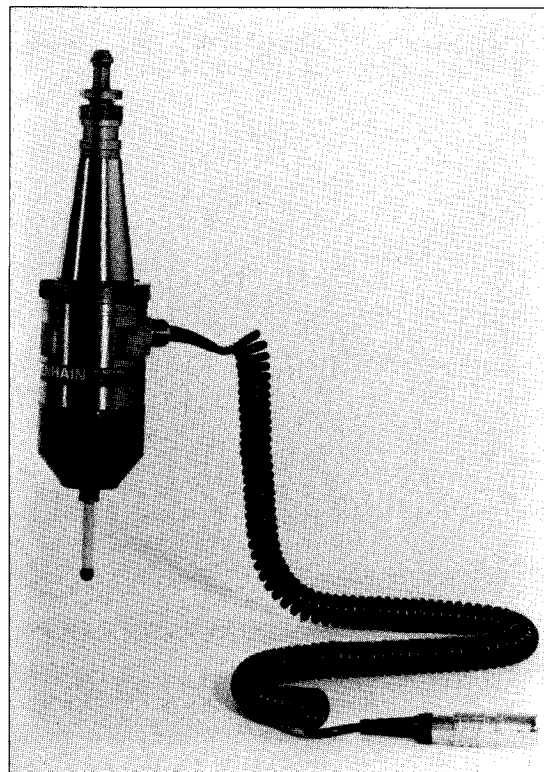
The probing functions allow compensation of workpiece misalignment and automatic setting of the position displays to help you setup workpieces more easily, quickly and accurately.

The probing functions can also be used for measurements on the workpiece.

Program run



You can program position measurements in the "Programming and editing" operating mode. This feature can be used with Q parameter programming to execute measurements before, during and after machining a piece (see index "Programming and Editing", Programmable probing function and Parameter programming).



TS 120

HEIDENHAIN offers touch probes in various versions. There are different clamping shafts to affix the probe head in the spindle like a tool. The stylus is replaceable. Standard versions are:

TS 120

Touch Probe System 120

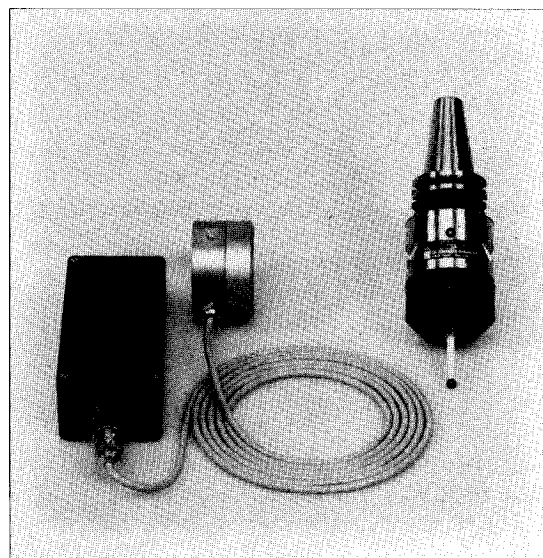
with cable connection and interface electronics, incorporated into probe.

TS 511

Touch Probe System 511

with infrared transmission, separate interface electronics and transmitter/receiver unit.

This probe head has a transmitter and receiver window (for the triggering signal) on one side and another transmitter window offset by 180°. The side with the transmitter and receiver window must be pointed towards the transmitter/receiver unit during measurement.



TS 511



Certain preparatory measures are required by the machine tool manufacturer for the connection of a touch probe system.

Accessories

FE 401 Floppy Disk Unit

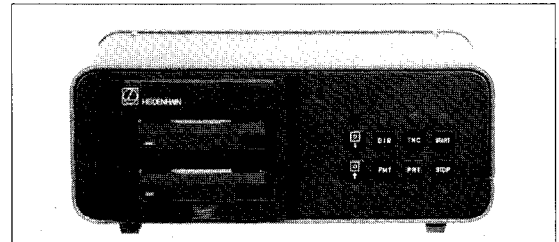
HR 130/HR 330 Electronic Handwheels

FE 401 Floppy Disk Unit

Part programs which do not have to reside permanently in the control memory can be stored with the FE 401 Floppy Disk Unit.

The storage medium is a normal 3 1/2 inch diskette, capable of storing up to 256 programs and a total of approximately 25000 program blocks.

Programs can be transferred from the TNC to diskette or vice-versa.



Programs written at off-line programming stations can also be stored on diskette with the FE 401 and read into the control as needed.

In the case of extremely long programs which exceed the storage capacity of the TNC, the FE 401 can be used to transfer a program blockwise into the control while simultaneously executing it.

A second diskette drive is provided for backing up stored programs and for copying purposes.

Specifications

	FE 401 Floppy Disk Unit with two drives
Data storage medium	3 1/2 inch diskette, double-sided, 135 TPI
Storage capacity	approx. 790 KB (25000 blocks); max. 256 programs
Data interface	Two RS-232-C data interfaces
Transfer rate	"TNC" interface: 2400/9600/19200/38400 baud "PRT" interface: 110/150/300/600/1200/2400/4800/9600/19200/38400 baud

Handwheel

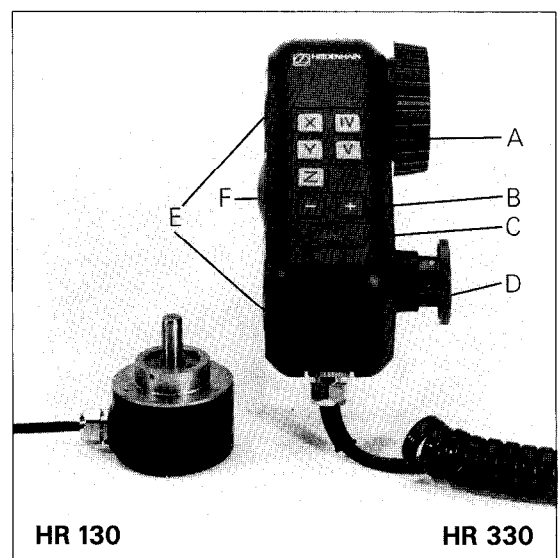
The control can be equipped with an electronic handwheel for better machine setup. Two versions of the electronic handwheel are available:

HR 130

Designed to be incorporated into the machine control unit. The axis of control is selected at the machine control panel.

HR 330

Includes keys for axis selection (A), axis direction (B), rapid traverse (C), emergency stop (D), magnetic holding pads (E) and enabling switch (F).





In addition to the main operating modes, the TNC has supplementary operating modes or so-called MOD functions. These permit additional displays and settings.

Initiate the dialog



Selecting

VACANT MEMORY 160044

Select MOD functions either via arrow keys or via the MOD key (only paging forward possible).

Terminating

LIMIT X+ = + 350.000

Terminate supplementary operating mode.

Transfer numerical inputs with the “ENT” key before terminating the MOD functions.

Vacant memory

The number of free characters in the program memory is displayed with the MOD function “VACANT MEMORY”.

Programming and editing

You can use this MOD function to switch the control between conversational format (**HEIDENHAIN**) and ISO format (**ISO**). Switchover is performed with the “ENT” key.

Baud rate

The transfer rate for the data interface is specified with “BAUD RATE”.

RS-232-C interface

The data interfaces can be switched via “RS-232-C interface” to the following operating modes with the “ENT” key:

- ME operation
- FE operation
- EXT operation: operation with other external devices.

NC software number

The software number of the TNC control is displayed with this MOD function.

PLC software number

The software number of the integrated PLC is displayed with this MOD function.

User parameters

Up to 16 machine parameters can be accessed by the machine operator with this MOD function. These user parameters are defined by the machine manufacturer – he may be contacted for more information.

Code number

A code number can be entered with this MOD function:

- **86357**: cancel “erase and edit protection”
- **123**: select the user parameters.

These user parameters are accessible on all controls (see User parameters).

MOD Functions

Position displays

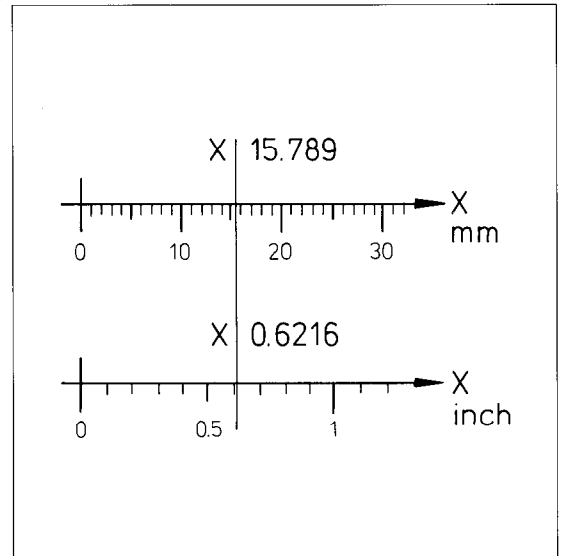
MOD

Change mm/inch

The MOD function "Change mm/inch" determines whether the control displays positions in the metric system (mm) or in the inch system. You switch between the mm and inch systems via the "ENT" key. After pressing this key the control switches to the other system.

You can recognize whether the control is displaying in mm or inches by the number of digits behind the decimal point:

X15.789 mm display
X 0.6216 inch display.



Position displays

The following position displays can be selected:

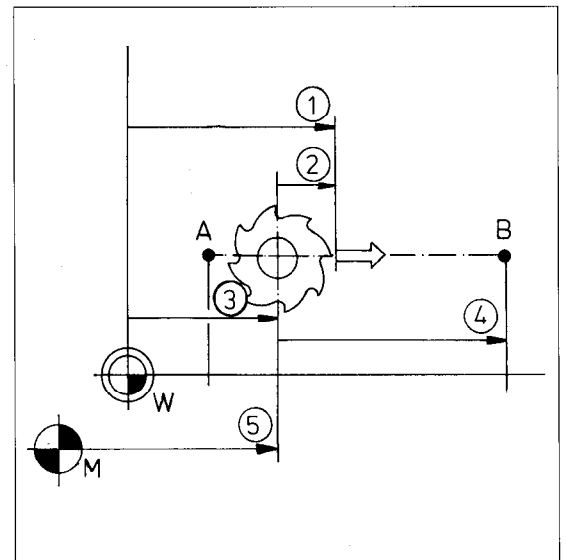
- | | |
|---|-------|
| ① nominal position of the control | NOML |
| ② difference nominal/actual position (lag distance) | LAG |
| ③ actual position | ACTL. |
| ④ remaining distance to programmed position | DIST. |
| ⑤ position based on the scale datum | REF |

A = last programmed position (starting position)

B = new (programmed) target position, which is presently targeted

W = Workpiece datum for the part program

M = scale datum (machine-based)



Switchover is with the "ENT" key.

Position display large/small

The character height of the position display can be changed in the operating modes "Program run/single block" or "Program run/full sequence". The position display shows 11 program blocks with small characters, two with large characters.

Switchover is with the "ENT" key.

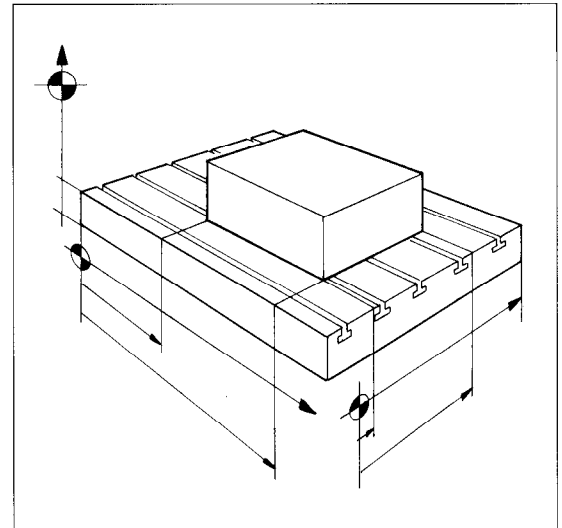
MOD Functions

Traverse range limits

MOD

Limits

The maximum displacements are preset by fixed software limits.
The MOD function "Limits" enables you to specify additional software limits for a "safety range" within the limits set by the fixed software limits. Thus you can, for example, protect against collision when clamping a dividing attachment. The displacements are limited on each axis successively in both directions based on the **scale datum** (reference marks). The position display must be switched to REF before specifying the limit positions of the position display.
To work without safety limits, enter the maximum values +30 000.000 or -30 000.000 for the corresponding axes.



 = scale datum

Effectiveness

The entered limits do not account for tool compensations. Like the software limit switches, they are only effective after you traverse the reference points. They are reactivated with the last entered values after a power interruption.

Determine values

To determine the input values, switch the position display to REF.



Traverse to the end positions of the axis/axes which is/are to be limited.
Note the appropriate REF displays (with signs).

Enter values

Select

  Continue pressing until LIMIT appears.

Enter the limit(s)

 Enter value, or

 select the next limit

...

 terminate the input.

User Parameters

General Information



Machine parameters

The TNC contouring controls are individualized and adapted to the machine via machine parameters (MP). These parameters consist of important data which determine the behavior and performance of the machine.

Parameters accessible for the user

Certain machine parameters which determine functions dealing only with operating procedures, programming and displays are accessible for the user.

Examples

- Scaling factor only effective on X, Y or on X, Y, Z.
- Adapting the data interface to different external devices.
- Display possibilities of the screen.

Accessibility

The user can access these machine parameters in two ways:

- Access by entering the **code number 123**.
This access is possible on every control (see code number 123).
- Access to additional parameters via the MOD function **User parameters**.
You can only access via the MOD function if the manufacturer has made the machine parameters accessible for this purpose.

The machine manufacturer can inform you about the sequence, meaning, texts etc. of any user parameters.

Only these machine parameters may be changed by the user. In no case should the user change any non-accessible machine parameters.

Selection



Select the user parameter.



Continue pressing until the desired
USER PARAMETER or dialog appears.



Enter numbers.



Terminate or select further
user parameters with  and
then terminate.

After entering the code number **123** via MOD, the following machine parameters and the parameters for the data interface (see index "Programming Modes", "External data transfer") can be selected and changed.

Measuring with the 3D touch probe

Function	Parameter no.	Input	Input values
Probe system selection	6010	0 → Cable transmission 1 → Infrared transmission	
Probe system: feed rate for probing	6120	80 to 3000 [mm/min]	
Probe system: measuring distance	6130	0 to 30 000.000 [mm]	
Probe system: set-up clearance over measuring point for automatic measurement	6140	0 to 30 000.000 [mm]	
Probe system: rapid traverse for probing	6150	80 to 29998 [mm/min]	

Display and programming

Function	Parameter no.	Input	Input values
Programming station	7210	0 → Control 1 → Programming station: PLC active 2 → Programming station: PLC inactive	
Block number increment	7220	0 to 255	
Switching of dialog language German/English	7230	0 → First dialog language 1 → Second dialog language (English)	
Inhibit PGM input for PGM no. = user cycle no.	7240	0 → Inhibited 1 → Uninhibited	
Central tool file	7260	0 → No central tool file 1 to 99 = Central tool file Input value = Number of tools	
Display of the current feed rate before start in the manual operating modes (same feed rate in all axes, i.e. smallest programmable feed rate)	7270	0 → No display 1 → Display	
Decimal character	7280	0 → Decimal comma 1 → Decimal point	
Display increment	7290	0 → 1 µm 1 → 5 µm	
Clearing the status display and the Q parameters with M02, M30 and end of program	7300	0 → Status display is not cleared 1 → Status display is cleared	
Graphics (display mode)	7310		
Switch over projection type "display in 3 planes"	Bit 0	+ 0 → Preferred German + 1 → Preferred American	
Rotate the coordinate system in the machining plane by 90°	1	+ 0 → No rotation + 2 → Coordinate system rotated by +90°	

**Machining and
program run**

Function	Parameter no.	Input	Input values
"Scaling" cycle is effective on 2 axes or 3 axes	7410	0 → 3 axes 1 → in the machining plane	
SL cycles for milling pockets with irregular contour	7420		
"Rough out" cycle: direction for pilot milling of contour	Bit 0	+ 0 → Pilot milling of contour for pockets counterclockwise, for islands clockwise + 1 → Pilot milling of contour for pockets clockwise, for islands counterclockwise	
"Rough out" cycle: sequence for rough out and pilot milling	1	+ 0 → First mill a channel around the contour, then rough out the pocket + 2 → First rough out the pocket, then mill a channel around the contour	
Joining compensated or uncompensated contours	2	+ 0 → Joining compensated contours + 4 → Joining uncompensated contours	
"Rough out" and "pilot milling" to pocket depth or for every infeed	3	+ 0 → "Rough out" and "pilot milling" are performed continuously over all infeeds + 8 → "Pilot milling" and then "rough out" are performed for every infeed (depending on bit 1) prior to the next infeed	
Overlap factor for pocket milling	7430	0.1 to 1.414	
Output of M functions	7440		
Programmed stop at M06	Bit 0	+ 0 → Programmed stop at M06 + 1 → No programmed stop at M06	
Output of M89, modal cycle call	1	+ 0 → No cycle call, normal output of M89 at start of block + 2 → Modal cycle call at end of block	
Constant path speed at corners	7460	0 to 179.999	
Display mode for rotary axis	7470	0 → 0 to 359.999 1 → ± 30 000.000	

Hardware

Function	Parameter no.	Input	Input values
Feed rate and spindle override	7620		
Feed rate override, if rapid traverse key is pressed in operating mode "Program run"	Bit 0	+ 0 → Override inactive + 1 → Override active	
Feed rate override in 2% increments or 1% increments	1	+ 0 → 2% increments + 2 → 1% increments	
Feed rate override, if rapid traverse key and external direction buttons are pressed	2	+ 0 → Override inactive + 4 → Override active	
Handwheel	7640	0 = Machine with electronic handwheel 1 = Machine without electronic handwheel	

Coordinates

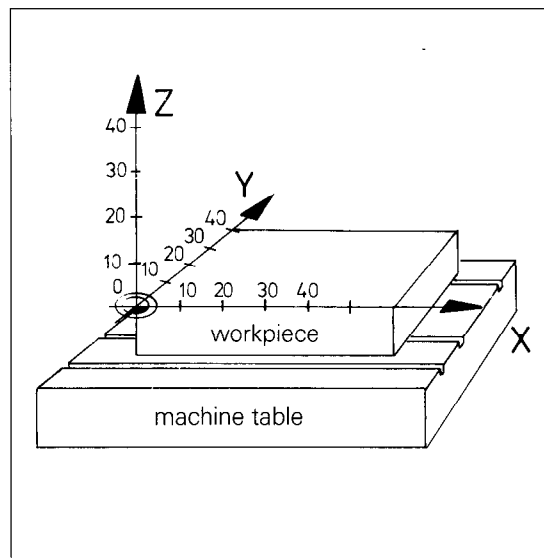
The coordinate system

In a part program, the **nominal positions** of the tool (or of the tool cutting edge) are defined in relation to the workpiece; encoders on the machine axes continuously deliver the signals needed by the control for determining the current **actual position**.

A reference system is always required for determining position. In the present case, such a system must be **workpiece-based**.

Cartesian coordinates

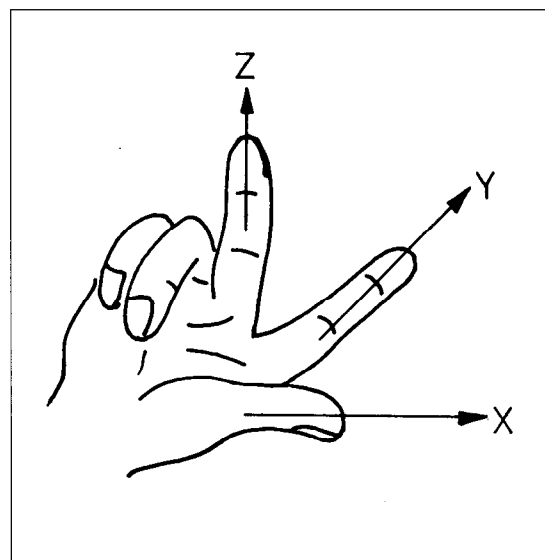
The reference system normally used is the **rectangular** or **Cartesian* coordinate system** (coordinates are those values which define a unique point in a reference system). The system consists of three coordinate axes, perpendicular to each other and lying parallel to the machine axes, which intersect each other at the so-called origin or (absolute) zero point. The coordinate axes represent mathematically ideal straight lines with divisions; the axes are termed X, Y and Z.



Right-hand rule

You can easily remember the traversing directions with the **right-hand rule**: the positive direction of the X axis is assigned to the thumb, that of the Y axis to the index finger, and that of the Z axis to the middle finger.

ISO 841 specifies that the **Z axis** should be defined according to the **direction of the tool spindle**, whereby the positive Z direction always points **from the workpiece to the tool**.



*) after the French mathematician René Descartes, in Latin Renatus Cartesius (1596 – 1650).

Coordinates

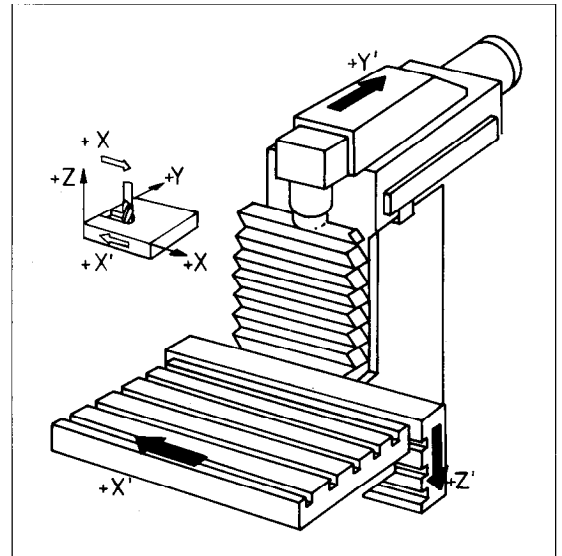
Datum

Relative tool movement

Part programs are always written with workpiece-based coordinates X, Y, Z. That is, they are written as if the tool moves and the workpiece remains still, independent of the machine type.

If, however, the work support on a given machine actually moves in any axis, then the direction of the coordinate axis and the direction of traverse will be opposite.

In such a case the machine axes are designated as X', Y' and Z'.



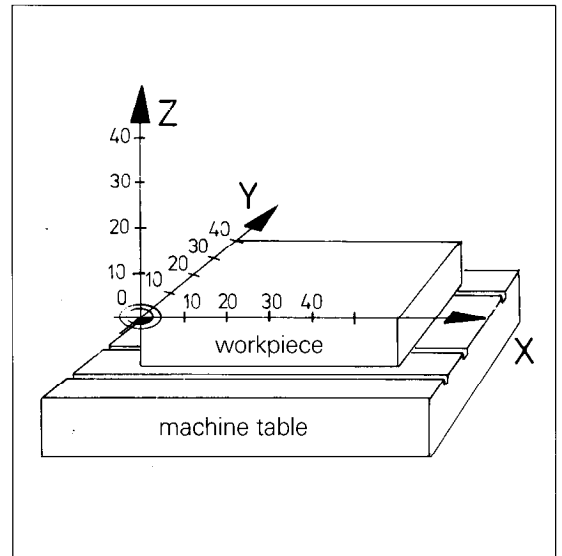
Zero point of the coordinate system

For the **zero point of the coordinate system**, the position on the workpiece which corresponds to the datum of the part drawing is generally chosen – that is, the point to which the part dimensioning is referenced.

For reasons of safety, the workpiece datum in the Z axis is almost always positioned at the highest point on the workpiece.

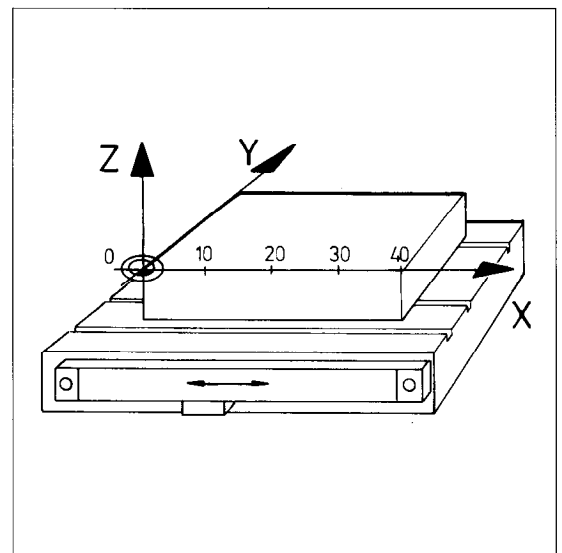
The datum position indicated in the drawing to the right is valid for all programming examples in this manual.

Machining operations in a horizontal plane require freedom of movement mainly in the positive X and Y directions. Infeeds starting from the upper edge of the workpiece $Z = 0$ correspond to negative position values.



Datum Setting

The workpiece-based rectangular **coordinate system** is defined when the coordinates of any **datum P** are known – that is, when the tool is moved to the datum position and the control “sets” the corresponding coordinates (datum setting).



Coordinates

Absolute and incremental coordinates

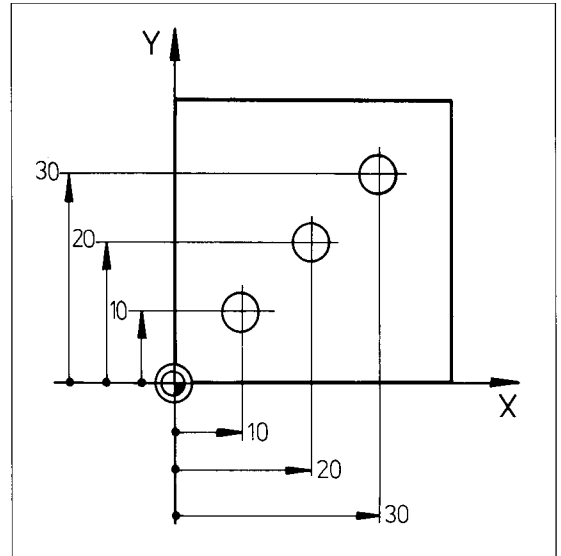
If a given point on the workpiece is referenced to the **datum**, then one speaks of **absolute coordinates** or absolute dimensions. It is also possible to indicate a position which is referenced to **another known workpiece position**: in this case one speaks of **incremental coordinates** or incremental dimensions.

Absolute dimensions

The machine is to be moved **to** a certain position or **to** certain nominal coordinates.

Example: G00 G90 X+30 Y+30

Dimensions in this manual are given as **absolute Cartesian dimensions** unless otherwise indicated.

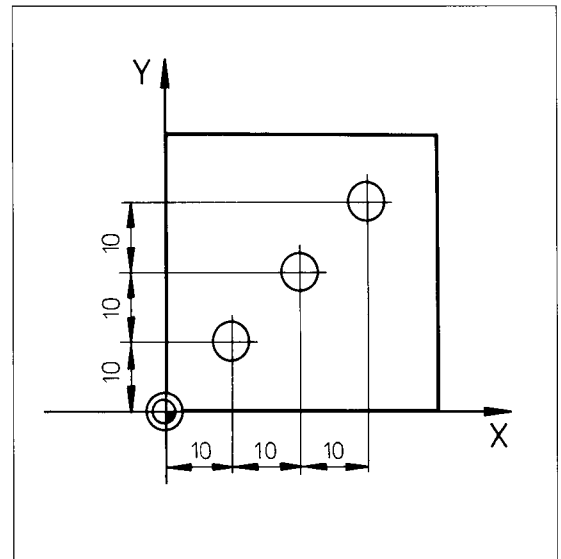


Incremental (chain) dimensions

Incremental dimensions in a part program always refer to the **immediately preceding nominal position**. Incremental dimensions are indicated by the letter I.

The machine is to be moved **by** a certain distance: it moves from the previous position along a distance given by the incremental nominal coordinate values.

Example: G00 G91 X+10 Y+10



Mixing absolute and incremental dimensions

It is possible to mix absolute and incremental coordinates within the same program block.

Example: G00 G91 X+10 G90 Y+30

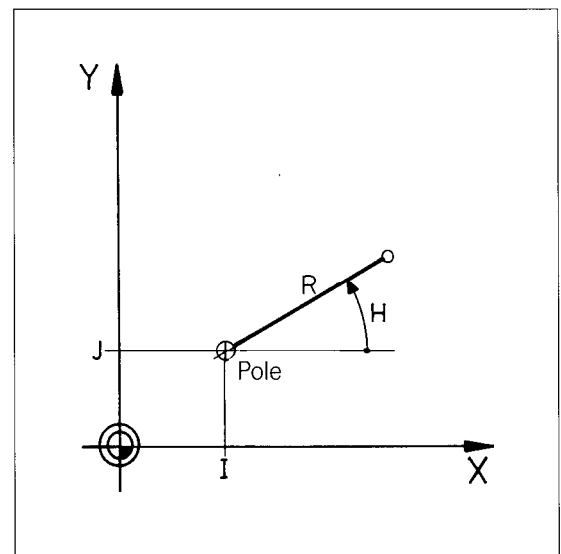
Polar coordinates

Positions on the workpiece can also be programmed by entering the radius and the direction angle referenced to a pole (see index Programming Modes, Polar coordinates).

I, J = Pole

R = Polar radius (distance from pole)

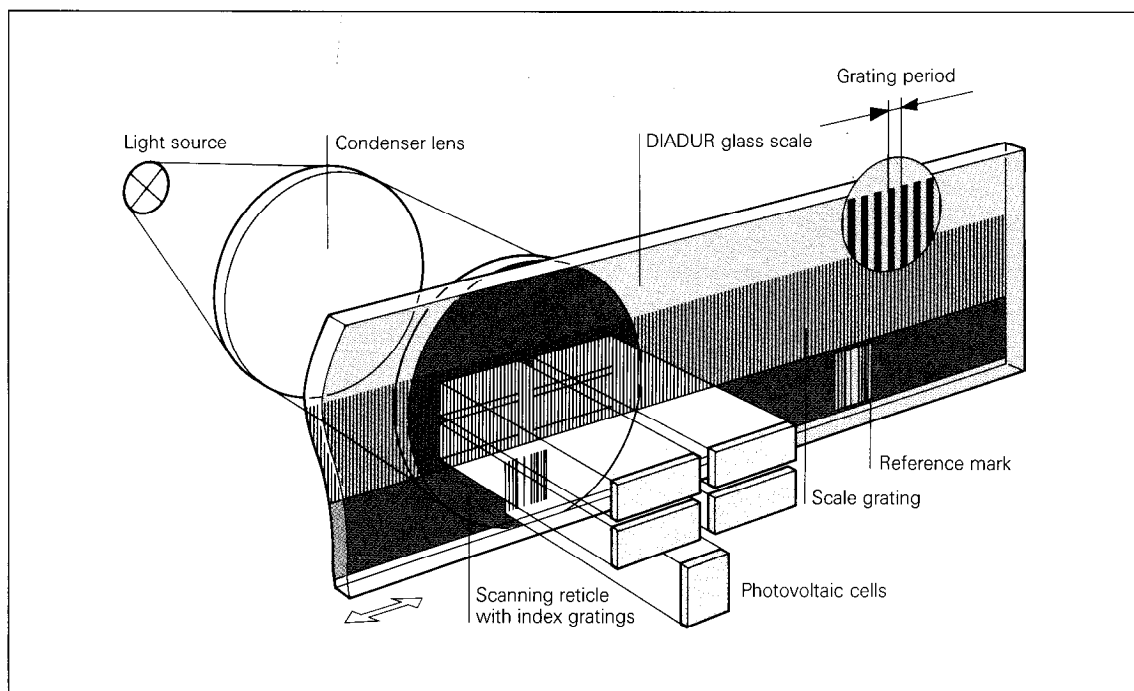
H = Polar angle (direction angle)



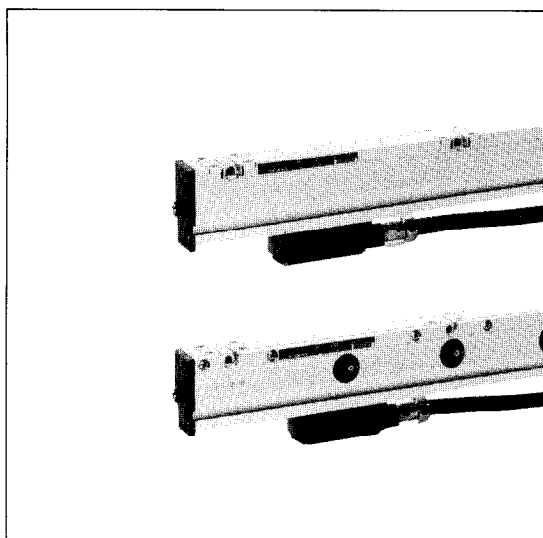
Linear and Angle Encoders

Linear and angle encoders in machine tools

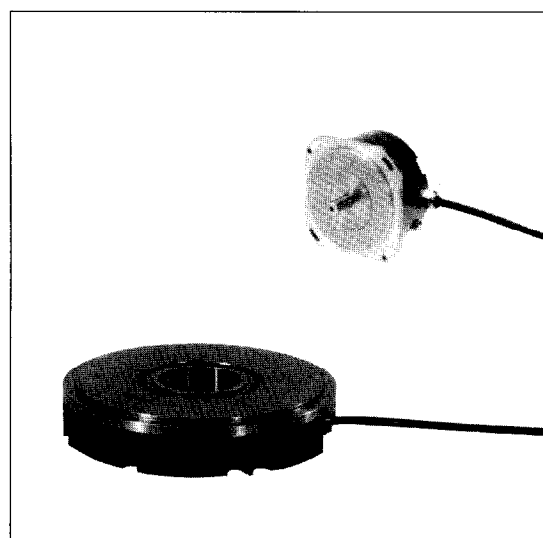
Each machine axis requires a measuring system to provide the control with information on the actual position: linear encoders for linear axes, angle encoders for rotary axes.



Principle of photoelectric scanning of fine gratings



LS 101C, LS 107C



RON 706C, ROD 250C

With **linear axes**, position measurement is generally based on either

- a photoelectrically scanned **steel or glass scale**, or
- the **high-precision ballscrew**, which also functions as the moving element (the electrical signals are then produced by a rotary encoder coupled to the ballscrew).

With **rotary axes**, a graduated disk permanently attached to the axis is photoelectrically scanned. The TNC forms the position value by counting the generated impulses.

Linear and Angle Encoders

Linear and angle encoders are **machine-based**:

Datum

The datum for determination of the nominal and actual position must correspond to the workpiece datum, or be brought into correspondence by **setting** the correct position value (= the position value determined by the workpiece datum) in any axis position. This procedure is called datum setting (or datum presetting).

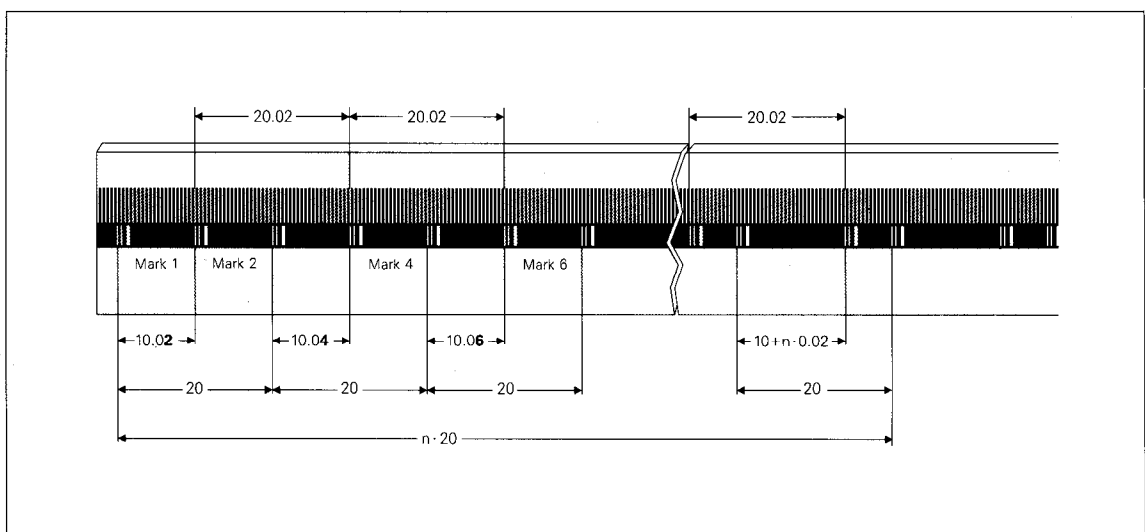
Reference marks

After the control has been switched off or after a power interruption, it is necessary to set the datum again. To simplify this task, the encoders possess **reference marks**, which in a sense also represent datum points.

The relationship between axis positions and position values which were established by the last setting of the workpiece datum (datum setting), are **automatically** retrieved by traversing over the reference marks after switch-on. This also re-establishes the machine-based references such as the software limit switch or tool change position.

In the case of linear encoders with distance-coded reference marks, the machine axes need only be traversed by a maximum of 20 mm. For angle encoders with distance-coded reference marks, a rotation of just 20° is required.

Linear encoders with only one reference mark have an **"RM"** label which indicates the position of the reference mark, while angle encoders with one reference mark indicate the position with a notch on the shaft.



Schematic of scale with distance-coded reference marks

Cutting Data

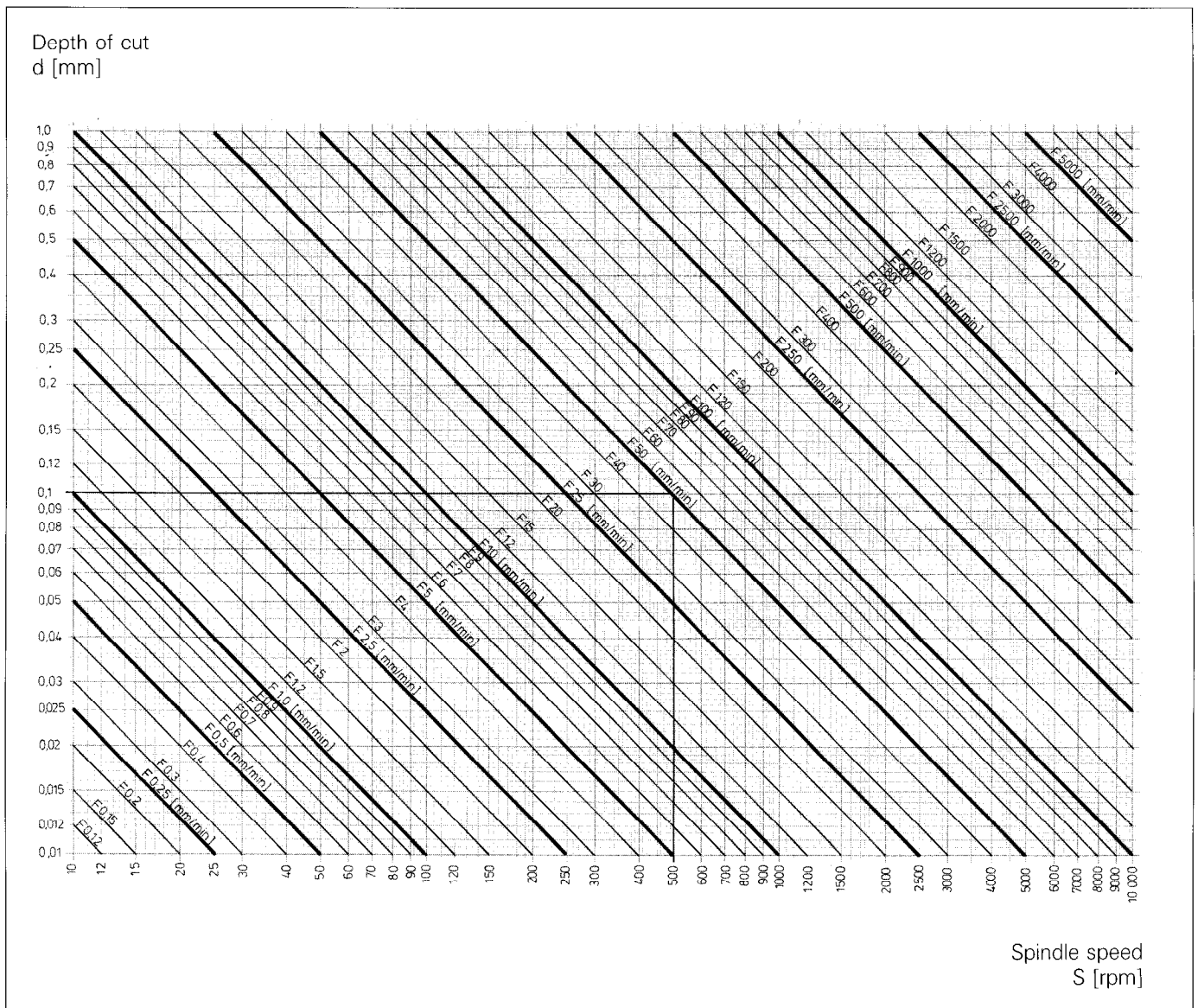
Feed rate diagram

The feed rate F must be defined in [mm/min] in the program. Usually, the number of teeth n on the tool, the permitted depth of cut d per tooth (in mm) and the previously determined spindle speed S (in rpm) are given. The diagram below helps you determine the feed rate F .

Determine the required **feed rate F** in [mm/min]

Given: n = number of teeth
 d = permitted depth of cut per tooth
 Selected: S = spindle speed
 Find: F = feed rate

Example
 6
 0.1 [mm]
 500 [rpm]



Calculation

Horizontal line through depth of cut 0.1 mm
 Vertical line through spindle speed 500 m/min
 At the point of intersection, read off the feed rate
 $F = 50$ mm/min; this is multiplied by the number
 of teeth $n = 6$:
 $F = 300$ mm/min

Formula

$$d = \frac{F}{S \cdot n} \text{ or } F = d \cdot S \cdot n$$

Prerequisites:

The feed rate determination assumes that

- the tool axis infeed = 1/2 tool radius
or
- the lateral infeed = 1/4 tool radius and the downfeed is selected equal to the tool radius.

Cutting Data

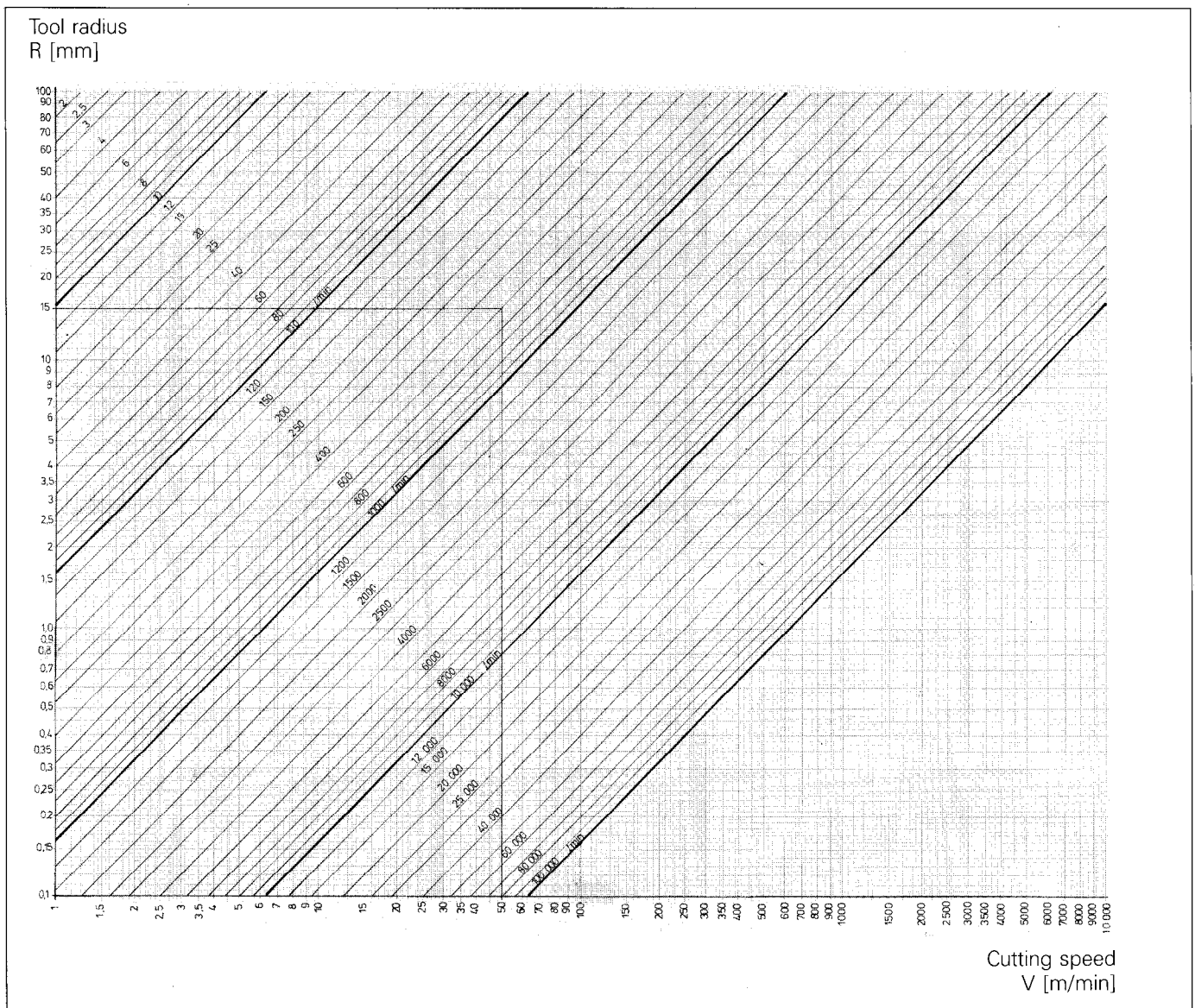
Spindle speed diagram

The spindle speed S must be entered in [rpm] in the part program. Usually the tool radius R is given in [mm] and the cutting speed V in [m/min]. The diagram below helps you determine the spindle speed S .

Determining the required **spindle speed S in [rpm]**

Given: R = tool radius
 V = cutting speed
 Find: S = spindle speed

Example
 16 [mm]
 50 [m/min]



Calculation

Horizontal line through the tool radius $R = 16$ mm
 Vertical line through the cutting speed $V = 50$ m/min

Read off the value at the point of intersection: approx. 500 rpm (calculated: 497 rpm)

Formula

$$V = 2R \cdot \pi \cdot S; \quad S = \frac{V}{2R \pi}$$

Cutting Data

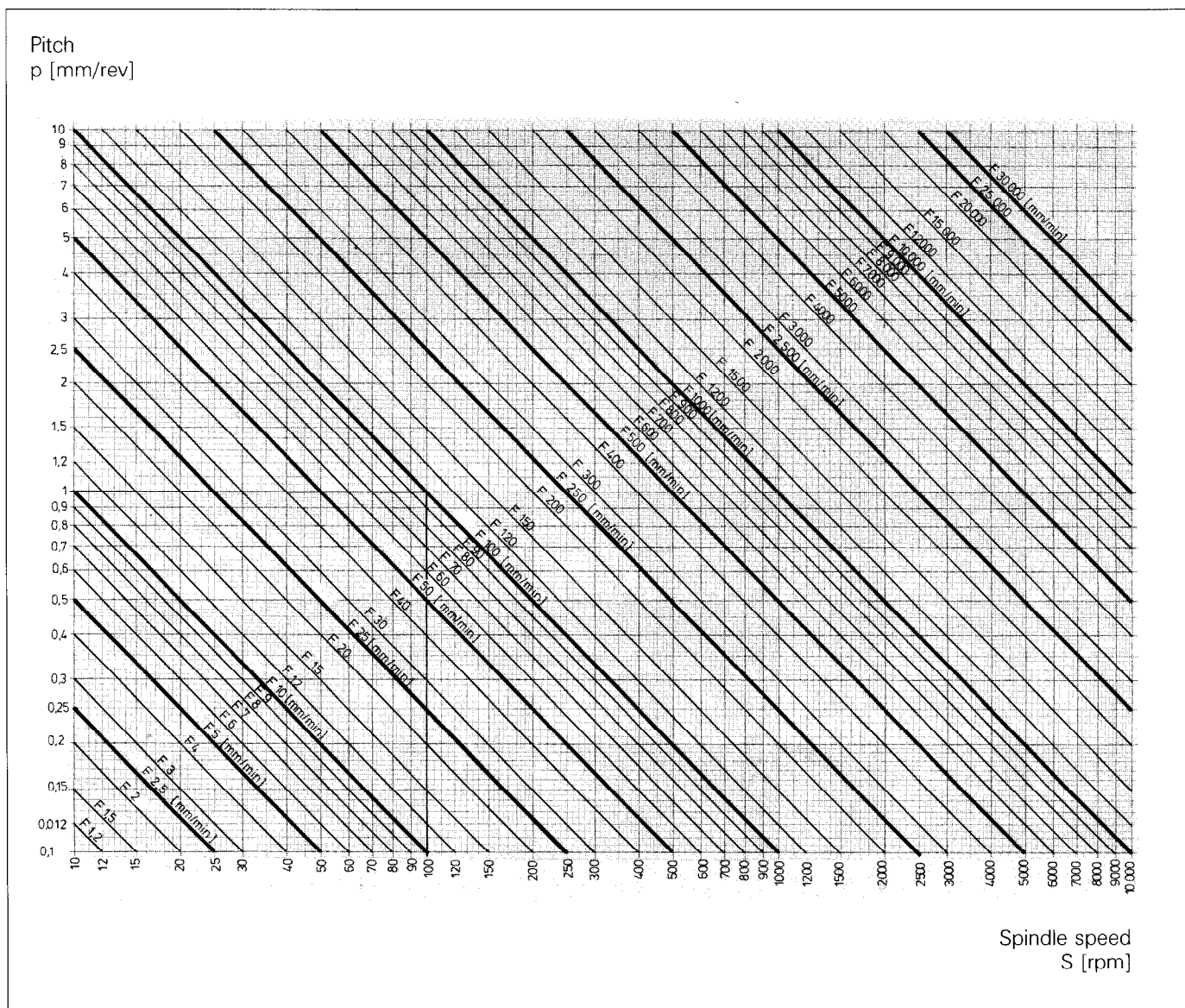
Feed rate diagram for tapping

When tapping a thread, the pitch P is given [mm/rev]. The spindle speed S and the feed rate F must be defined in the program. First, the spindle speed is determined in the appropriate diagram, then the feed rate is found in the diagram below.

Determine the required **feed rate F** in [mm/min]

Given: p = pitch [mm/rev]
 Selected: S = spindle speed [rpm]
 Find: F = feed rate [mm/min]

Example
 1 [mm/rev]
 100 [rpm]



Calculation

Horizontal line through pitch $p = 1.0$ mm/rev
 Vertical line through spindle speed $S = 100$ rpm

Read off feed rate at point of intersection:
 $F = 100$ mm/min to tap this thread.

Formula

$$p = \frac{F}{S} \text{ or } F = p \cdot S$$

Machine Operating Modes (M)



Switch-On

Traversing the reference points 1

Manual Operation

Traversing with the axis direction buttons 2

Spindle speed S/Miscellaneous functions M 2

3D Touch Probe

or



Datum setting with probe system 3

Calibrating effective length 4

Calibrating effective radius 5

Reference surface, Position measurement 6

Basic rotation, Angular measurement 7

Corner = datum/Determining corner coordinates 9

Circle center = datum/Determining the circle radius 11

Datum setting without probe system

13

Electronic Handwheel/ Incremental Jog

15

Positioning with Manual Data Input (MDI)

Tool call/Spindle axis/Spindle speed 17

Positioning to entered coordinates 18

Program Run

Single block, Full sequence 19

Interrupting the program run 20

Checking/changing Q parameters 21

Background programming 22

Blockwise transfer (drip feed) 23

Switch-On

Traversing the reference points



Switch-On

 Switch power on.	
MEMORY TEST	The TNC tests the internal control electronics. The display is automatically cleared.
POWER INTERRUPTED	 Delete the message. The control then tests the EMERGENCY STOP circuit.
RELAY EXT. DC VOLTAGE MISSING	 Switch on the control DC voltage.
MANUAL OPERATION TRAVERSE REFERENCE POINTS Z AXIS X AXIS Y AXIS 4th AXIS	 Traverse the axes over the reference points in the displayed sequence. Start each axis separately or  ... move the axes with the external direction keys. The sequence of the axes is determined by the machine manufacturer.
MANUAL OPERATION	"Manual operation" is now selected automatically.

Encoders

The required traversing distance for linear and angle encoders with distance-coded reference marks is max. 10 mm or 20 mm/10° or 20°.
If the encoder has only one reference mark, it must be traversed.

Manual Operation

Traversing with the axis direction buttons/ Spindle speed S/Miscellaneous functions M



The machine axes can be moved and the datum set in the "Manual" operating mode.

Jog mode

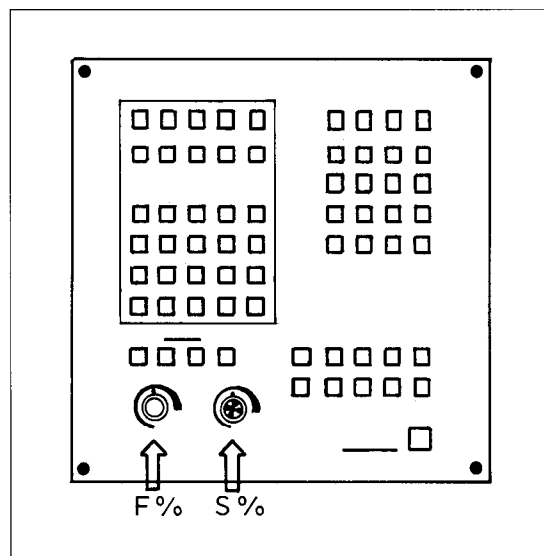


The machine axis moves as long as the corresponding external axis direction button is held down. Several axes can be driven simultaneously in the jog mode.

Continuous operation



If the machine "START" button is pressed simultaneously with an axis direction button, the selected machine axis continues to move after the two buttons are released. Movement is stopped with the external "STOP" button.



Feed rate override

The traverse speed (feed rate) is preset by machine parameters and can be varied with the feed rate override (F %) of the control.

Note

If a block number increment between 1 and 255 is selected (see index M "General Information", user parameter MP 7220), the block number can be omitted since it is generated automatically by pressing a function key.

Spindle speed



Enter the block number.



Enter spindle speed S (e.g. 1000).

Example

N10 S 1000 *

Spindle override

On machines with continuously variable spindle drives, the speed can also be varied with the spindle override (S %).

Miscellaneous function



Enter the block number.



Enter the M function (e.g. M03).

Example

N10 M03 *

Combination

It is also possible to enter both spindle speed and miscellaneous function M in one block.

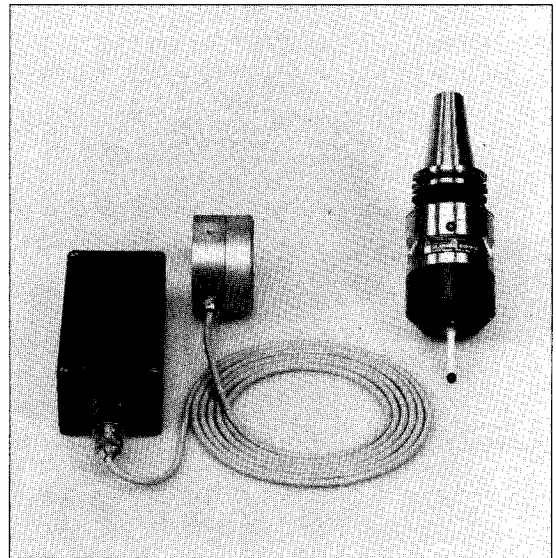
Example

N10 S1000 M03 *



Using the touch probe for setup

For workpiece setup the 3D touch probe systems from HEIDENHAIN in association with TNC software offer considerable benefits. One is that the workpiece does not have to be aligned precisely to the machine axes: The TNC will determine and compensate misalignment automatically ("basic rotation"). Another important benefit of the 3D touch probe systems is significantly faster and more accurate datum setting.



TS 511

Probing functions



The touch probe functions described below can also be employed in the "electronic handwheel" operating mode.

Pressing the "TOUCH PROBE" key calls the menu shown here to the right. The probing function is selected with the cursor keys and entered with the "ENT" key.

CALIBRATION EFFECTIVE LENGTH
CALIBRATION EFFECTIVE RADIUS
BASIC ROTATION
SURFACE = DATUM
CORNER = DATUM
CIRCLE CENTER = DATUM

Calibration

The effective length of the probe and the effective radius of the probing ball must be calibrated once, before beginning touch probe work. Both dimensions are determined by CALIBRATION routines and stored in the control.

Terminating the probing functions

The probing functions can be terminated at any time with "END □".

Process

The probe head traverses to the side or upper surface of the work. The feed rate during measurement and the maximum measuring distance are set by the machine manufacturer via machine parameters.

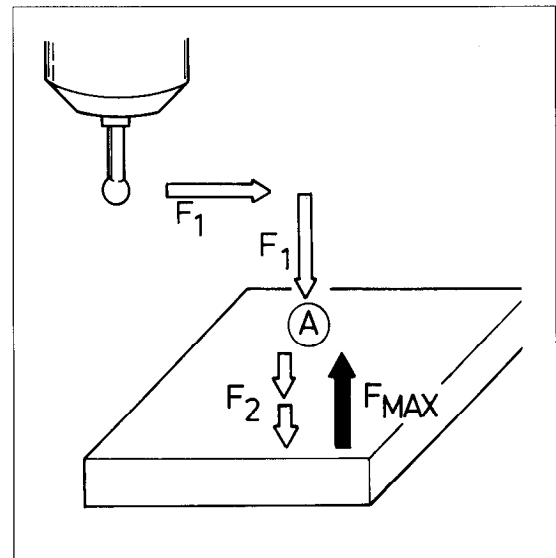
The touch probe system signals contact with the workpiece to the control. The control stores the coordinates of the contacted points. The probing axis is stopped and retracted to the starting point. Overrun caused by braking does not affect the measured result.

Ⓐ = pre-positioning with the external axis direction buttons.

F1 = feed rate for pre-positioning.

F2 = feed rate for probing.

FMAX = retraction in rapid traverse.



3D Touch Probe

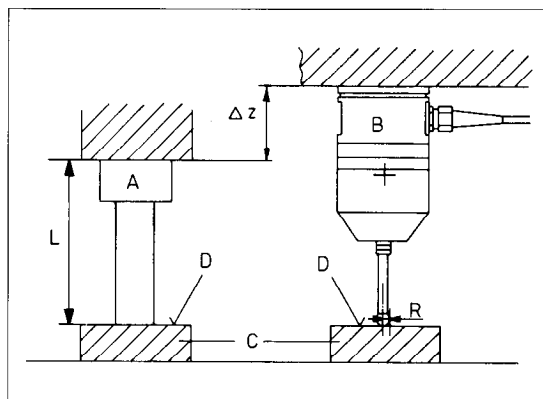
Calibrating effective length



Work aid: ring gauge

For calibration of the effective length, a ring gauge of known height and known internal radius is clamped to the machine table.

- A = zero tool
- B = 3D touch probe
- C = ring gauge
- D = reference plane (surface)
- L = length of the zero tool
- R = ball tip radius
- ΔZ = effective length of probe system

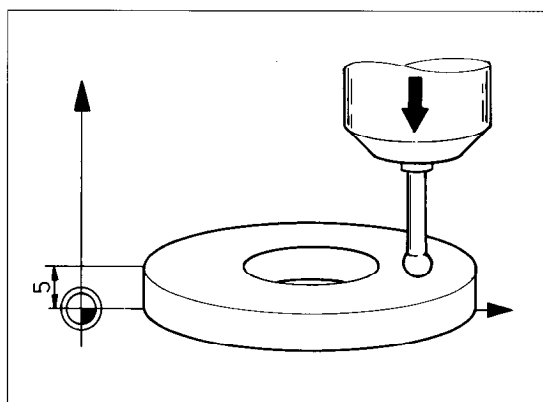


Procedure

The reference plane is set with the zero tool prior to calibration.

To determine the effective length of the stylus, the probe head touches the reference plane. After contacting the surface, the probe head is retracted in rapid traverse to the starting position.

The length L is stored by the control and automatically compensated during the measurements.



Initiate the dialog



CALIBRATION EFFECTIVE LENGTH



Select probing function and enter.

TOOL AXIS = Z

☐ Enter a different tool axis if required.



Select the "Datum".

DATUM +5

☐ Enter the datum in the tool axis, e.g. +5.0 mm.

Z+ Z-



Move the touch probe to the vicinity of the reference plane.



Select the direction of probe movement, here Z-.



The probe head moves in negative Z direction.

After touching the surface and returning to the starting position, the control automatically switches to the "Manual operation" or "Handwheel" operating mode.

Display

The value for effective length can be displayed by selecting "Calibration effective length" again.

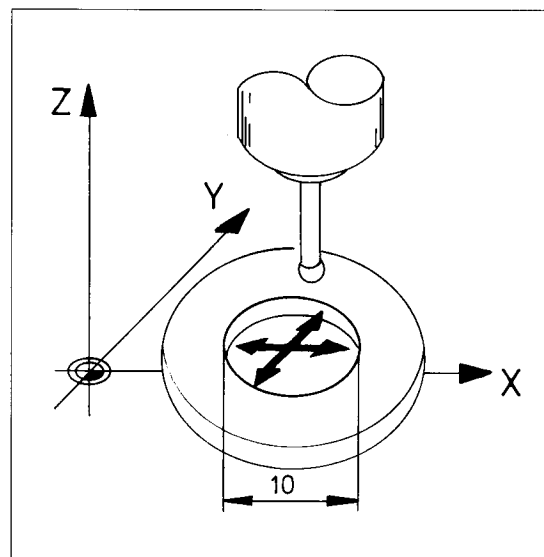


Procedure

The probe ball is lowered into the bore of the ring gauge. 4 points on the wall must be touched to determine the effective radius of the stylus ball. The traverse directions are determined by the control, e.g. X+, X-, Y+, Y- (tool axis = Z).

The probe head is retracted in rapid traverse to the starting position after every deflection.

The radius R is stored by the control and automatically compensated during the measurements.



Initiate the dialog



CALIBRATION EFFECTIVE RADIUS



Select probing function and enter.

TOOL AXIS = Z



Enter another tool axis if required.



Select "Radius ring gauge".

RADIUS RING GAUGE = 10



Enter the radius of the ring gauge, e.g. 10.0 mm.

X+ X- Y+ Y-



Traverse approximately to the center of the ring gauge.



Select the traversing direction of the probe head (only necessary if you prefer a certain sequence or the exclusion of one probing direction).



Probe a total of 4 times. After contacting the wall of the ring gauge four times, the control automatically switches to the "Manual operation" or "Handwheel" operating modes.

Display

You can display the value for effective radius by selecting "Calibration effective radius" again.

Error messages

All touch probe systems:

TOUCH POINT INACCESSIBLE

The stylus was not deflected within the measuring distance (machine parameter).

STYLUS ALREADY IN CONTACT

The stylus was already deflected at the start.

Touch probe system TS 511:

PROBE SYSTEM NOT READY

Probe system not set up correctly, or transmission path was interrupted.

The transmitter and receiver window (i.e. the side with two windows) must be pointed towards the transmitter/receiver unit.

3D Touch Probe

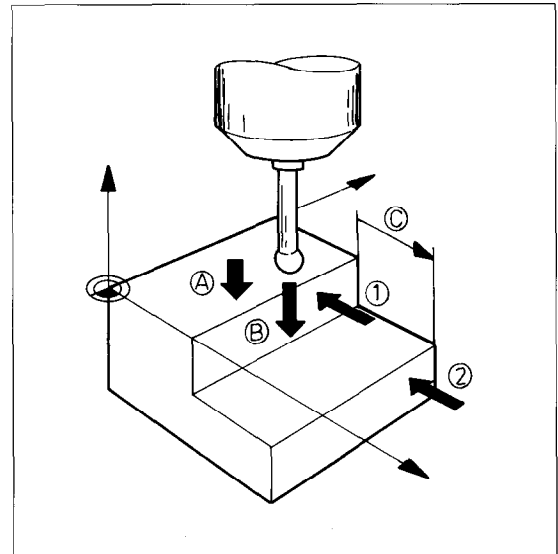
Reference surface, Position measurement



The position of a surface on the clamped work-piece is determined with the probing function "Surface = datum".

Functions

- Setting the reference plane Ⓐ
- Measuring positions Ⓑ
- Measuring distances Ⓒ



Measuring positions

Initiate the dialog

SURFACE = DATUM



Select probing function and enter.

X+ X- Y+ Y- Z+ Z- C+ C-



Move to the starting position.



Select the traversing direction, e.g. Z-.



Move the probe head in negative Z direction. The probe head is retracted in rapid traverse to the starting position after touching the surface.

Measured value

DATUM Z-18.125

The control displays the measured value.

Setting the reference plane

DATUM Z+0



Enter a new value if required, e.g. 0 mm.



Confirm entry.

Measuring distances

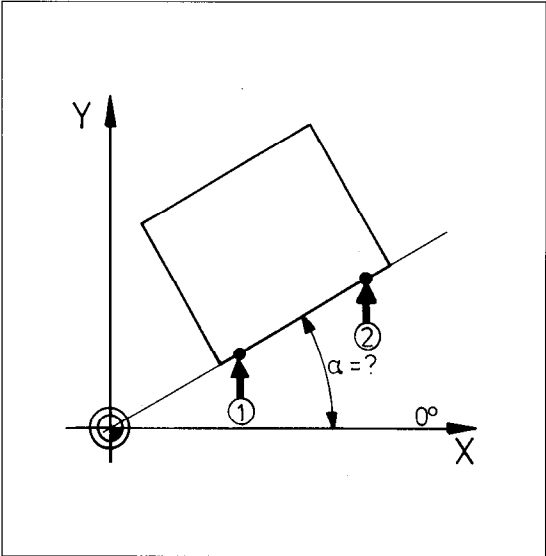
You can also measure distances on an aligned workpiece.

- Probe the first position and set the datum (e.g. 0 mm).
- Probe the second position.
The distance can be read in the "Datum" display.

The probing function “Basic rotation” determines the angle of deviation of a plane surface from a nominal direction. The angle is determined in the machining plane.

Functions

- Basic rotation (the control compensates for an angular misalignment)
- Correct an angular misalignment (on a machine tool with rotary axis)
- Measure an angle.



Basic rotation

Initiate the dialog



BASIC ROTATION



Select probing function and enter.

ROTATION ANGLE = 0



Select the “Rotation angle”.



Enter the nominal direction of the surface to be probed, e.g. 0°.

X+ X- Y+ Y-



Move the probe head to the starting position ①.



Select the probing direction, e.g. Y+.



The probe head travels in the selected direction, e.g. Y+.

The probe head returns to the starting position after touching the side surface.



Move the probe head to the starting position ②.



The probe head travels in the selected direction, e.g. Y+.

The probe head returns to the second starting position after making contact. The control automatically switches to the “Manual operation” or “Handwheel” operating mode.

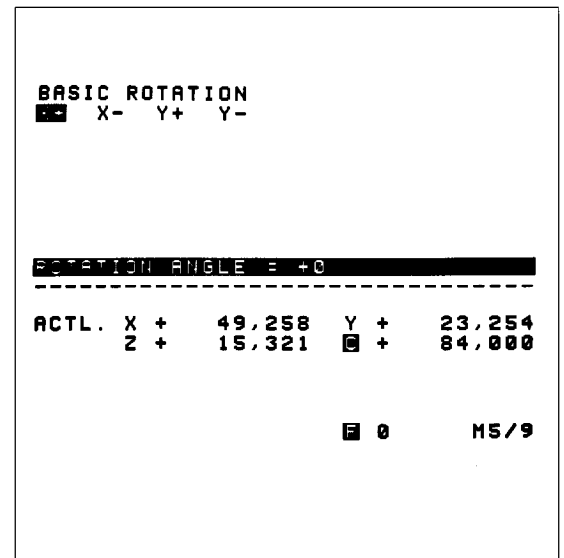


Displaying the rotation angle



The measured rotation angle is displayed by selecting the probing function "Basic rotation".

Compensation of angular misalignment is registered on the screen with "ROT" in the status display. It also remains stored after a power interruption.



Cancelling the basic rotation (rotation angle 0°)



The basic rotation is cancelled by selecting the probing function "Basic rotation" and entering a 0° rotation angle. The "ROT" display is cleared.

Once basic rotation is activated, all subsequent programs are executed with rotation and shown rotated in the graphic simulation.

Measuring angles

In addition to basic rotation, angle measurements can also be performed on aligned workpieces.

Carry out the following procedure:

- Execute a basic rotation.
- Display the rotation angle.
- Cancel the basic rotation.

Compensating for misalignment

On machine tools with a rotary axis, you can also correct misalignment of a workpiece by rotating the axis.

Carry out the following procedure:

- Execute a basic rotation.
- Display and note the rotation angle.
- Cancel the basic rotation.
- Enter the noted value for the rotary axis **incrementally** in the "Positioning with MDI" operating mode (see "Positioning to entered position") and start the rotation with the machine "START" button.

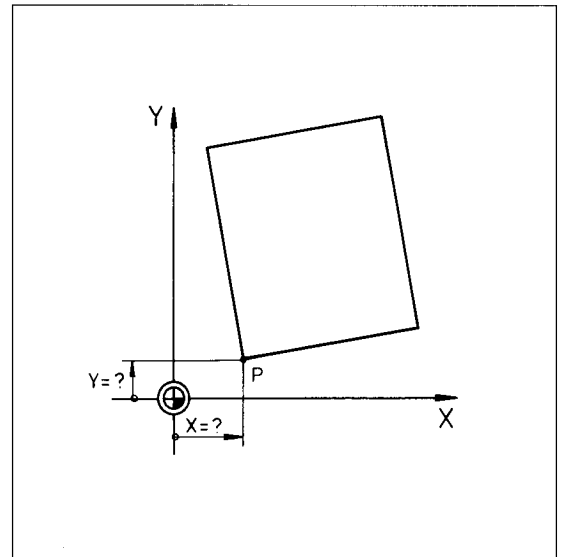
3D Touch Probe

Corner = datum/ Determining corner coordinates



With the probing function "Corner = datum", the control computes the coordinates of a corner on the clamped workpiece. The computed value can be taken as datum for subsequent machining. All nominal positions then refer to this point.

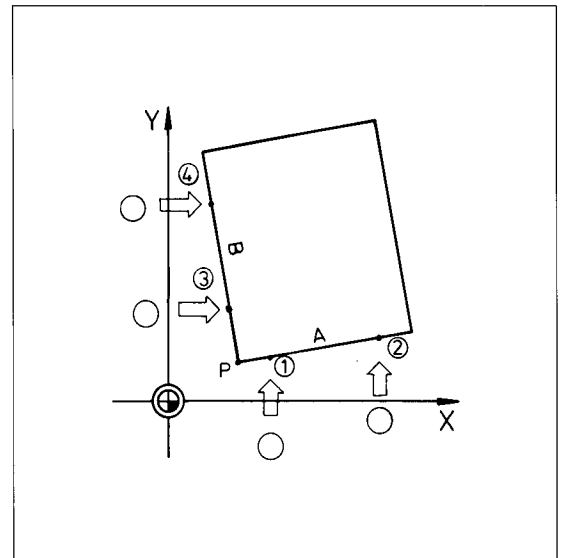
The probing function "Basic rotation" should be performed before "Corner = datum".



Process

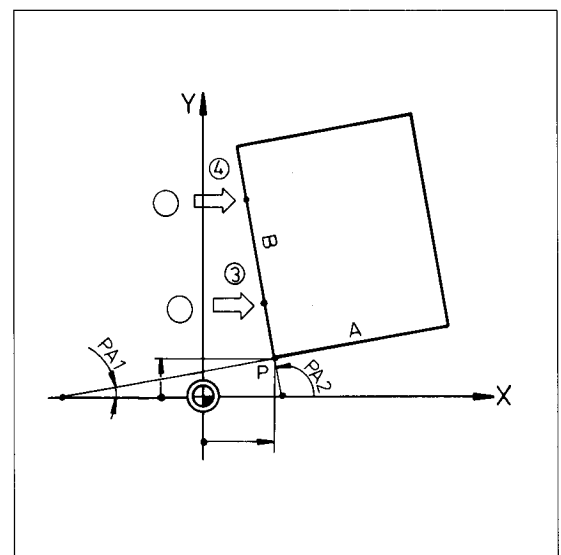
The probe head touches two side surfaces (see figure) from two different starting positions per side.

The corner point P is computed by the control as the intersection of straight line A (contact points ① and ②) with straight line B (contact points ③ and ④).



After performing a basic rotation

If the probing function "Corner = datum" is called after performing a basic rotation (straight line A), the first side need not be contacted.



3D Touch Probe

Corner = datum/ Determining corner coordinates



To transfer the direction of the first side face from the "basic rotation" routine, simply respond to the dialog query **TOUCH POINTS OF BASIC ROTATION ?** by pressing the "ENT" key (otherwise "NO ENT").



If only the probing function "CORNER = DATUM" is performed, then it does not contain a basic rotation.

Initiate the dialog



CORNER = DATUM



Select probing function and enter.

First side face

X+ X- Y+ Y-



Move the probe head to the first starting position.



Select the probing direction, e.g. Y+.



The probe head travels in the selected direction.

After touching the side face, the probe head is retracted to the starting position.

Traverse to the second starting position and probe in the same probing direction as described above.

Second side face

X+ X- Y+ Y-



Move the probe head to the third starting position.



Select the probing direction, e.g. X+.



The probe head travels in the selected direction.

After touching the side face, the probe head is retracted to the starting position.

Traverse to the fourth starting position and probe in the same probing direction as described above.

**Display corner coordinates/
Setting the datum**

DATUM X+0

DATUM Y+0



Enter the corner coordinates for X and Y if required, e.g. X+0, Y+0.



Confirm entries.

3D Touch Probe

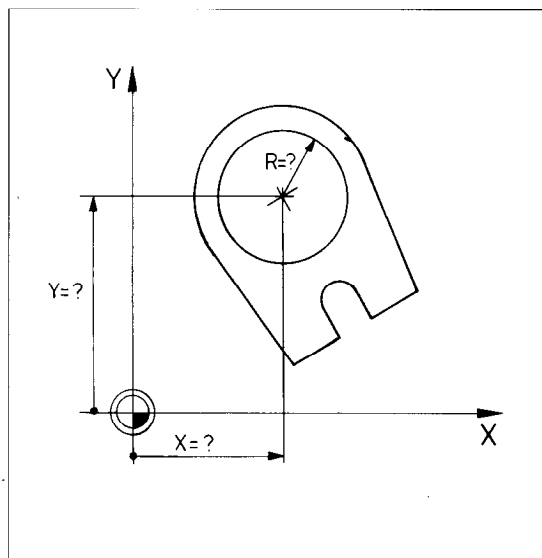
Circle center = datum/ Determining the circle radius



In the probing function "Circle center = datum", the control computes the coordinates of the circle center and the circle radius on a clamped workpiece with cylindrical surfaces. The coordinates of the center can be used as the datum for subsequent machining. All nominal positions are then referenced to this point.

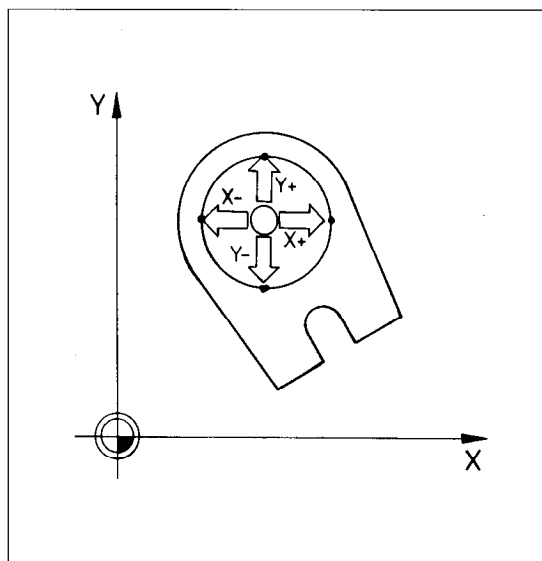


The "Basic rotation" probing function must be carried out prior to "Circle center = datum".



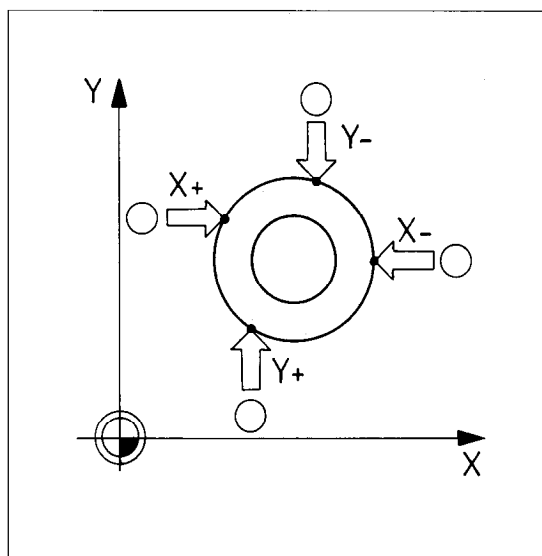
Bore, Circular pocket

Position the probe head in the bore with the remote axis direction keys. 4 different positions are then touched by pressing the machine START button.



Outer cylinder

On workpieces with cylindrical outer surfaces, the probing directions must be specified for each of the four points.



3D Touch Probe

Circle center = datum/ Determining the circle radius



Initiate the dialog



CIRCLE CENTER = DATUM



Select the probing function
and enter.

X+ X- Y+ Y-



Move the probe head to the first
starting position.



Select the probing direction if required,
e.g. X-.



Probe head travels in the selected
direction.

After touching face, the probe head is
retracted to the starting position.

Traverse to the second and third starting positions
and probe in different directions as described
above.

X+ X- Y+ Y-



Move the probe head to the fourth
starting position.



Select the probing direction if required,
e.g. Y-.



The probe head travels in the selected
direction.

The probe head is retracted to the starting
position after touching the side face.

Display

X+54.3 Y+21.576

Coordinates of the circle center.

PR+20

Circle radius.

Datum setting

DATUM X+40



Enter the X and Y coordinates of the circle
center if necessary, e.g. X+40, Y+30.

DATUM Y+30



Confirm entries.

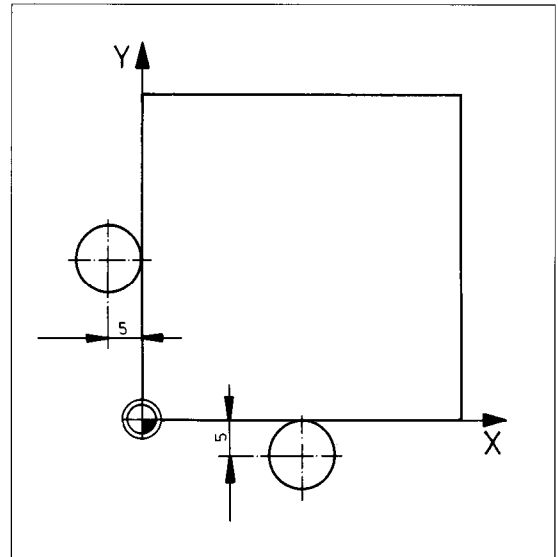


Align workpiece and set datum

First align the workpiece parallel to the machine axes in the conventional way. For datum setting the machine is then moved to a known position relative to the workpiece and the relevant position values are entered with the axis keys.

Touching in the working plane

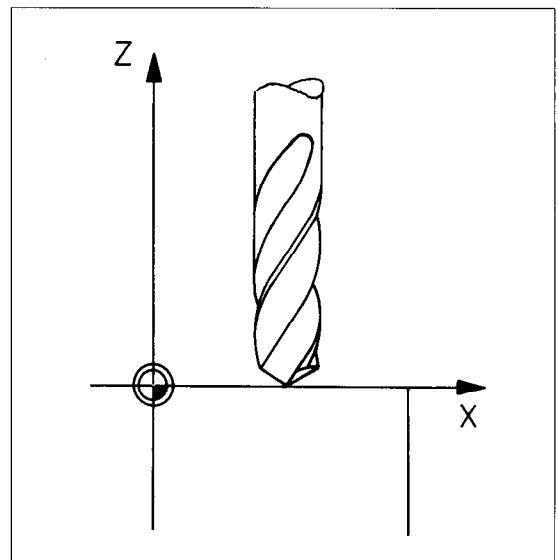
Touch both sides of the workpiece with a tool or edge finder and, at contact, set the actual position display of the associated axis to the tool radius or the ball tip radius of the edge finder with a **negative** sign (here e.g. $X = -5 \text{ mm}$, $Y = -5 \text{ mm}$).



Touching in the tool axis (spindle axis)

The actual position display is set to zero when the zero tool touches the work surface.

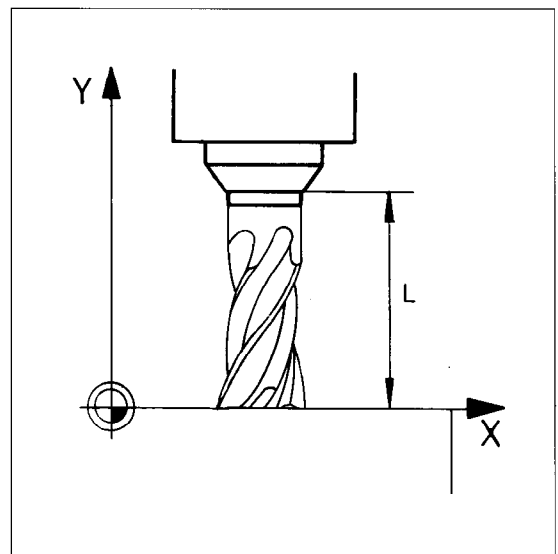
If the workpiece surface must not be scratched, you can lay a metal shim of known thickness (e.g. 0.1 mm) on it. Then enter the thickness of the shim when contact is made (e.g. $Z = +0.1 \text{ mm}$).



Preset tools

When using preset tools (i.e. when the tool lengths are already known) touch the work surface with any tool. To assign the value 0 to the surface, enter the length L of the inserted tool with a positive sign as the actual value for the infeed axis. If the work surface has a value other than 0, enter the following actual value:
 $(\text{actual value } Z) = (\text{tool length } L) + (\text{surface position})$

Example:
 tool length L : 100 mm
 position of the work surface: +50 mm
 actual value to be entered:
 $Z = 100 \text{ mm} + 50 \text{ mm} = 150 \text{ mm}$



Manual Operation

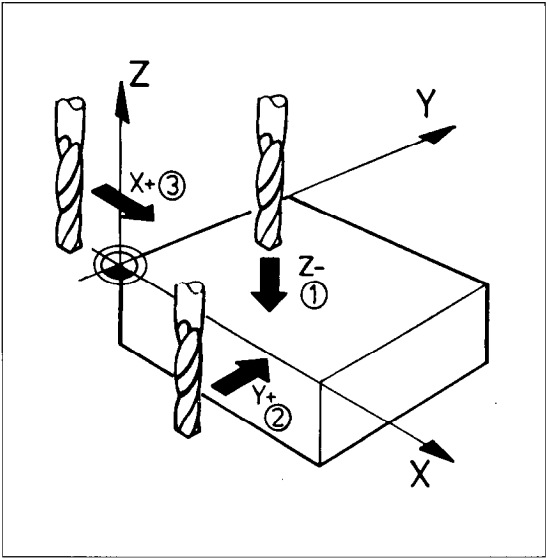
Datum setting without probe system



Example: Setting the datum

The datum is to be set with a drill (tool radius $R = 5 \text{ mm}$) as shown to the right.

- ① Touch the workpiece surface.
- ② Touch side by moving the Y axis.
- ③ Touch side by moving the X axis.



Touching with Z axis

Initiate the dialog

Z, after surface ① is touched.

DATUM SET Z =



Enter the value for the Z axis, e.g. 0 mm.



Confirm entry.

The Z display reads: 0.000

Y axis

Initiate the dialog

Y, after surface ② is touched.

DATUM SET Y =



Enter the value for the Y axis, e.g. 5 mm.



Here with a negative sign.



Confirm entry.

The Y display reads: -5.000

X axis

Initiate the dialog

X, after surface ③ is touched.

DATUM SET X =



Enter the value for the X axis, e.g. 5 mm.



Here with a negative sign.



Confirm entry.

The X display reads: -5.000

The datum for the fourth axis can be set in a similar way.

If the dialog **DATUM SET** was opened by mistake, the dialog can be terminated with "NO ENT" or "END □".

Active datum points are only shown in the "ACTUAL" position display.
This display may have to be selected with "MOD" (see index A "General Information/MOD Functions - Position displays").

Electronic Handwheel/Incremental Jog



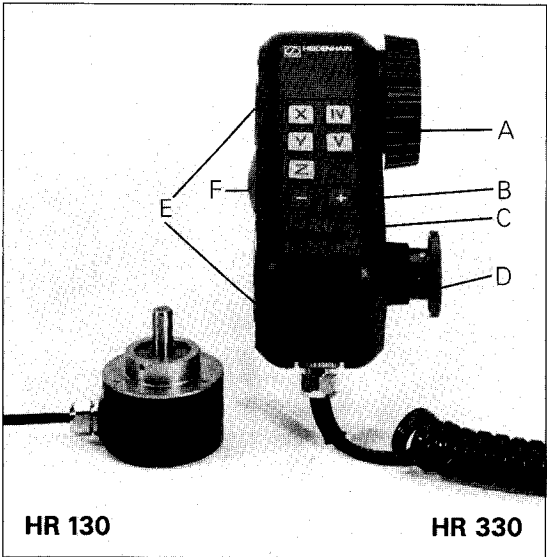
Versions

The control is usually equipped with an electronic handwheel. It can be used, for example, to set up the machine.

There are two versions of the electronic handwheel:

HR 130: to be incorporated into machine operating panel

HR 330: portable version with
axis selection keys (A),
axis direction keys (B),
rapid traverse key (C),
EMERGENCY STOP button (D),
magnetic holding pads (E),
enabling switch (F).



Interpolation factor

The displacement per handwheel turn is determined by the interpolation factor (see table to the right).

Interpolation factor	Displacement in mm per turn
0	20.0
1	10.0
2	5.0
3	2.5
4	1.25
5	0.625
6	0.313
7	0.156
8	0.078
9	0.039
10	0.020

Operating the HR 130

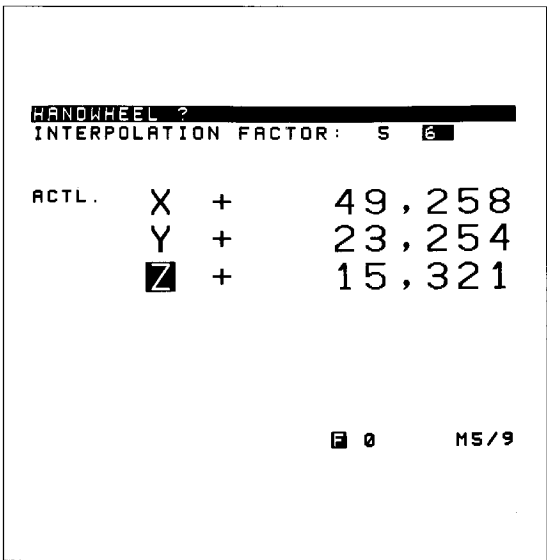
The handwheel is switched to the required machine axis with the axis keys of the control.

Operating the HR 330

The axis is selected on the handwheel.
The axis to be driven by the electronic handwheel is highlighted in the screen display.



In the "Electronic handwheel" operating mode, the machine axes can also be driven with the external axis direction buttons.





Operating the
HR 130/330

Set operating mode and initiate the dialog



INTERPOLATION FACTOR: 3



Enter the desired interpolation factor,
e.g. 4.



Confirm entry.

INTERPOLATION FACTOR: 4



Select the axis:
on the control (HR 130)
or on the handwheel (HR 330)

The tool can now be moved in a positive
or negative Y direction with the electronic hand-
wheel.

Incremental jog
positioning

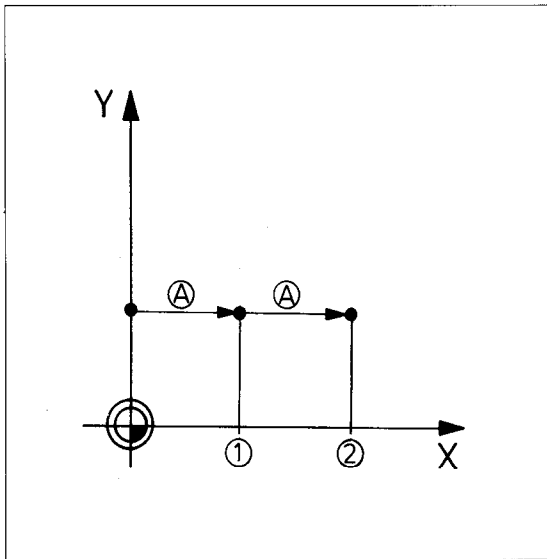
The machine manufacturer can activate incre-
mental jog positioning via the integral PLC. In this
case, a traversing increment can be entered in
this operating mode.

The axis is moved by the entered increment
when you press a machine axis button. This can
be repeated as often as desired. Only single-axis
movements are possible.

Ⓐ Jog increment: e.g. 2 mm.

① Machine axis button (e.g. X) pressed once.

② Machine axis button pressed twice.



Entering the
jog increment

Set operating mode and initiate the dialog



JOG-INCREMENT: 1.000



Enter the jog increment, e.g. 2 mm.



Confirm the entry.

JOG-INCREMENT: 2.000



or another remote axis key.

The axis is driven by the entered jog increment.

Positioning with Manual Data Input (MDI)

Tool call/Spindle axis/Spindle speed



A tool must first be defined before tool radius compensation can be called with G41/G42 in the “Positioning with MDI” mode of operation. A tool can be defined either in the central tool file or within a part program.

If no central tool file is used, you must define the tool with G99 in the “Program run, single block” or “Program run, full sequence” mode.

The significance of “G99” and “T” are explained in index P “Programming Modes, Tool Definition”.

**Example:
tool call**

Input



Tool number



Select spindle axis. e.g. Z



Spindle speed



Conclude block



Tool call

Positioning with Manual Data Input (MDI)

Positioning to entered coordinates



In the "Positioning with MDI" mode, paraxial positioning blocks (i.e. for traverse in only one axis) can be entered and executed. The entered blocks are not saved in memory.

Approaching the position

Input



Paraxial positioning

			No radius compensation or
			Paraxial compensation for increased length (R+) or
			Reduced length (R-)
			Absolute or
			Incremental dimensions
			Axis and coordinate value, e.g. X axis
			Feed rate
			M function
			Conclude block



Start positioning block.

Terminate block entry



Terminates block immediately. Earlier entries for tool radius compensation, feed rate, direction of spindle rotation remain effective.

Paraxial radius compensation

For paraxial positioning blocks you need only enter whether the tool path is shortened or lengthened by the tool radius.

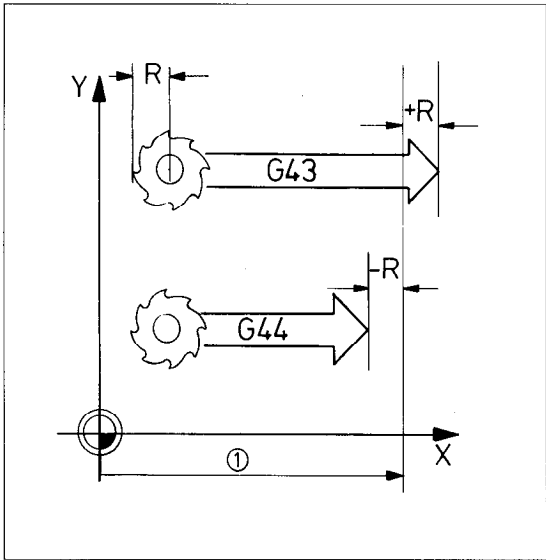
G43 lengthens the tool path.

G44 shortens the tool path.

If a G43/G44 radius compensation is also entered for the angular positioning of the spindle axis it will be ignored.

Nor is a tool radius compensation effective for a fourth axis when used for a rotary table.

① Nominal position



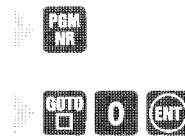
The workpiece datum must be set before machining the work!
See "Datum setting with/without probe system".

Program run single block is best used for program test and for the first program run.

Operating mode



Single block



Select the program or, if the program was already selected:

```
select block 0.
```

0 BEGIN PGM 7225

The first program block is shown in the current line of the program.



Each program block must be started with the machine START button.

You must restart the program to continue after a programmed stop.

Operating mode



Full sequence

Select the program and block number as described above.



The program runs continuously up to a programmed stop or end of program.

The programmed feed rate can be varied via the feed rate override.

The programmed spindle speed can be varied via the spindle override (if output is analog).

Program Run

Interrupting the program run

Stop



Stop program run:
Stop axis movements with the machine STOP button.
The block currently being processed is not completed.
The "control in operation" (*) display blinks.

Abort



Interrupt program run.
The "control in operation" (*) display is cleared.

The control stores:

- the last tool called
- coordinate transformations
- the last valid circle center/pol CC
- the current program section repeat
- the return jump label for subprograms

Switching to single block



In the "Program run, full sequence" operating mode, you can interrupt the program run by switching to "Single block".

The block currently being executed is completed.

Program run is to be discontinued after execution of the current block.



To continue, either start each block separately or reactivate "Program run, full sequence".

EMERGENCY STOP

The machine can be switched off in an emergency by hitting one of the EMERGENCY STOP buttons.

The control acknowledges this with the message
EMERGENCY STOP

To continue working, release the emergency stop key (usually by turning it clockwise), then

1. Remove the cause of error
2. Switch on the control power again
3. Clear the message EMERGENCY STOP with the "CE" key
4. Restart the program run.

Program Run

Checking/changing Q parameters



Q parameters

**Interrupt
program run**

You can check and, if necessary, change Q parameters after interrupting the program run.



Stop program run by pressing the machine STOP button.
Interrupt program run.

**Check
parameter**



Select and check the desired parameter.

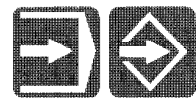
**Change
parameter**



Terminate Q parameter display or
change the parameter and confirm.

Program Run

Background programming



Programming during program execution

The control permits the execution of a program in the “program run, full sequence” mode **at the same time** as **another** program is being edited, graphically tested or transferred via RS-232-C (V.24) or RS-422 (V.11) interface in the “programming and editing” mode. This parallel operation is especially useful for transferring data or making small program changes while running long programs which require little attention from the operator.

A program cannot be run and edited at the same time.

Starting the part program

Operating mode



Initiate the dialog



PROGRAM NUMBER =

Select part program.



Start machining.

Parallel operating mode: programming and editing

Operating mode



Select and edit the program

or

transfer a program via the RS-232-C/V.24 data interface.

Screen display

The screen is divided into two halves during parallel operation: The program to be edited is shown in the upper half. The program currently in process appears in the lower half: program number, current block number and current status are displayed.

Terminating the parallel operating mode

Operating mode



Parallel operating is terminated by pressing the “Program run, full sequence” key.

Program Run

Blockwise transfer

(drip feed)



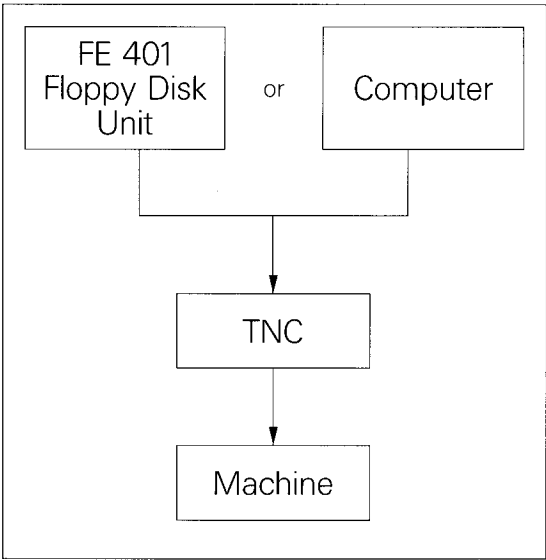
Execution from external storage

In the “Program run, full sequence” or “Single block” operating mode, part programs can be “transferred blockwise” from a remote computer, a storage medium or a HEIDENHAIN FE unit via the RS-232-C/V.24 serial data interface. Thus allows execution of part programs which exceed the storage capacity of the control.

Data interface

The data interface is programmable via machine parameters (see index “Programming Modes”, External data transfer).

The RS-232-C interface of the TNC must be set for external transfer or FE operation!



Program structure

Only programs without jumps can be executed with “Blockwise transfer”.

- Program calls, subprogram calls, program section repeats and conditional program jumps cannot be executed.
- Unless the control operates with a central tool file, only the tool last defined can be called.

Block numbers (sequence numbers)

The program to be transferred can have block numbers (sequence numbers) exceeding 999. The blocks do not have to be numbered sequentially; however, no block number may exceed the number 65534. High sequence numbers are displayed with 2 lines.

Starting “blockwise transfer”



Data transfer from an external storage device can be started in the operating modes “Program run, full sequence/single block” with the “EXT” key.

The control stores the transferred program blocks in available memory and interrupts data transfer when the storage capacity is full.

No program blocks are displayed until the available memory is full or the program is completely transferred.

The program run can be started with the machine “START” button even when no program block is displayed.

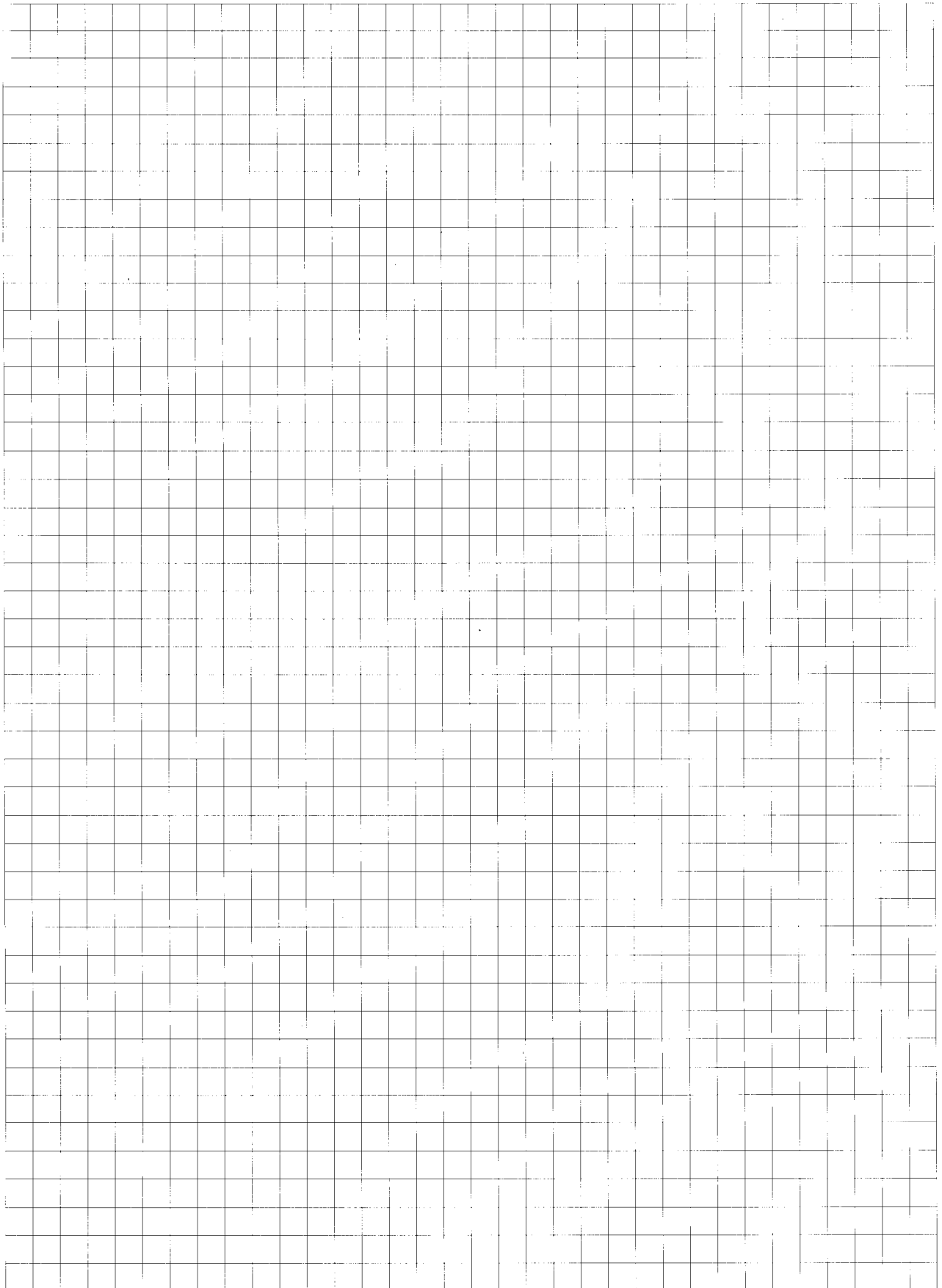
To avoid unnecessary interruptions of the program run, you should already have a number of stored program blocks as a buffer before starting. Therefore, it is advantageous to wait until the available memory is full.

After starting, the executed blocks are discarded and further blocks are continuously called from the external storage device.

Skipping over program blocks



If, in “Blockwise transfer” operation, you press the “GOTO □” key before starting and enter a sequence number, all blocks preceding this number will be ignored.





Programming in ISO

Fundamentals	1
Sequence numbers/Block format	2
Editing functions	3
Clearing/deleting functions	5



Program Selection

Opening a program		6
Erase/edit protection	G50	7
Defining the workpiece blank	G30/G31	8

Tool Definition



Tool definition within the part program	G99	10
Tool definition in program O		11
Transferring tool length		13
Tool radius		14

Cutter Path Compensation

Entering the radius compensation	G41/G42	15
Working with radius compensation		16
Radius compensation	G43/G44	17

Tools

Tool call		18
Tool change		19

Feed rate F/ Spindle Speed S/ Miscellaneous Function M Programmable STOP/ Dwell Time

	20
G38	21

Path Movements

Input	22
Overview of path functions	23
1D/2D/3D movements	24

Linear Movement, Cartesian

Positioning in rapid traverse	G00	25
Drilling	G01	26
Chamfer	G24	27
Example		28
Additional axes		29

Circular Movement, Cartesian

Interpolation planes	30
Selection guide: Arbitrary transitions	31
Tangential transitions	32



Circular Movement, Cartesian

Arc with circle center I, J, K	G02/G03	33
Arc with radius	G02/G03	35
Corner rounding with radius R	G25	37
Tangential arc with end point X/Y	G06	39

Polar Coordinaten

Fundamentals		41
Pole	I, J, K	42
Straight lines	G10/G11	43
Circular arcs	G12/G13/G15	44
Tangential arcs	G16	45
Corner rounding RND	G25	45
Helical interpolation with poles I, J, K	G12/G13	46

Contour Approach and Departure

Starting and end position		48
On a circle with radius R	G26/G27	50

Predetermined M Functions

Constant contour speed:	M90	51
Small contour steps:	M97	52
Terminating compensation:	M98	53
Machine-referenced coordinates:	M91/M92	54

Program Jumps

Overview		55
----------	--	----

Jumps Within a Program

Program labels	G98	56
Program section repeats		57
Subprograms		59
Nesting subprograms		61
Example: Hole pattern with several tools		62
Example: Horizontal geometric form		63

Program Calls

64



Standard Cycles

Introduction, Overview 65

Fixed cycles

Preparatory measures		66
Pecking	G83	67
Tapping with floating tap holder	G84	70
Slot milling	G74	71
Rectangular pocket milling	G75/G76	73
Circular pocket milling	G77/G78	75

SL cycles

Fundamentals		77
Contour geometry	G37	78
Rough-out	G57	78
Roughing-out a rectangular pocket		80
Roughing-out a rectangular island		81
Overlaps		82
Overlapping pockets		83
Overlapping islands		86
Overlapping pockets and islands		87
Pilot drilling	G56	89
Contour milling (finishing)	G58/G59	90
Machining with several tools		91

Coordinate Transformations

Overview		93
Datum shift	G54	94
Mirror image	G28	96
Coordinate system rotation	G73	98
Scaling	G72	100

Other Cycles

Dwell time	G04	102
Program call	G39	103
Oriented spindle stop	G36	104

Parametric Programming

Overview		105
Selection		106
Algebraic functions		107
Trigonometric functions		108
Conditional/unconditional jumps		110
Special functions		111
Example: Bolt hole circle		113
Drilling with chip breaking		114
Ellipse as an SL cycle		115
Sphere		117



Programmed Probing		
	Overview	G55 120
	Example: Measuring length and angle	121
	Teach-In	123
	Test Run	125
	Graphic Simulation	126
GRAPHICS		
		
	External Data Transfer	
	General information	129
	Transfer menu	130
	Connecting cable/Pin assignment for RS-232-C	131
	Peripheral devices	132
	FE floppy disk unit	133
	Non-HEIDENHAIN devices	134
	Machine parameters	135
Address letters in ISO		137

Programming in ISO Fundamentals



Introduction

The individual work steps on a conventional machine tool must be initiated by the operator. On an NC machine, the numerical control assumes computation of the tool path, coordination of the feed movements on the machine slides and generally also monitors the spindle speed. The control receives the information for this in form of a program in which the machining of the workpiece is described.

This program can be considered a work plan.

"Programming" means creating and entering a work plan in a form which is understood by the control.

Program start and specification of blank

Define and call a tool, move to the tool change position.

Move to the workpiece contour, machine the workpiece contour, depart from the workpiece contour,

Traverse to the tool change position.

End of program

Program scheme

Programs

The control can store up to 32 programs (HEIDENHAIN or ISO) with a total of 4000 blocks (HEIDENHAIN dialog).

One part program can contain up to 1000 blocks. Individual programs are identified by program numbers.

Switching between conversational and ISO programming

The control is switched to conversational or ISO programming via the MOD functions (see index A "General Information, MOD functions, Programming and editing").

Program input

Once the control has been switched from conversational to ISO programming, the functions of the keys correspond to the snap-on keyboard.

The control "STOP" key is covered by the "D" key. In ISO programming, the "DEL" key assumes the function of the "STOP" key.

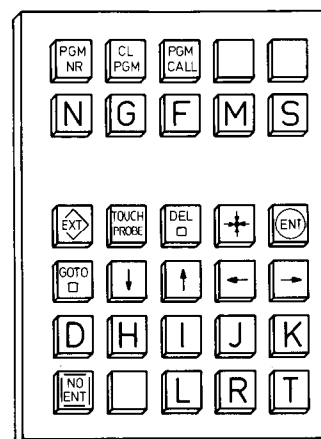
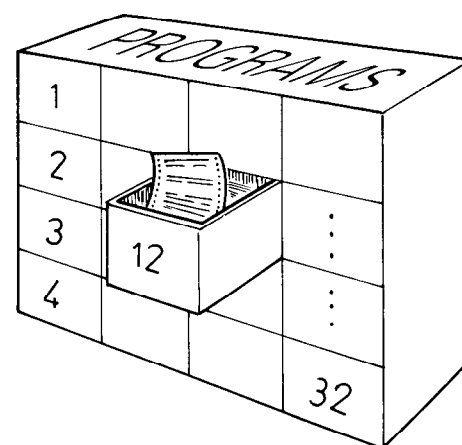
ISO programming is partly dialog guided. The individual commands (words), except for the dimensional data (G90, G91), can be entered in any sequence within a block. The commands are then sorted after the block has been concluded.

Program beginning

At the beginning of an ISO program, the control requires information on:

- The working plane (G17/G18/G19)
- Programming of absolute/incremental dimensions (G90/G91)
- Radius compensation (G40/G41/G42)

The first positioning block should look like this:
G00 G90 G40 G17 Z+200



Programming in ISO

Sequence number/Block format



Sequence number

The sequence number identifies the program block in a part program. If a sequence number increment between 1 and 255 is set in the machine parameter MP 7220 (see index A "General Information, User parameters") the sequence number will be generated automatically, eliminating the need to enter each sequence number by hand.

The numerical sequence of block numbering has no effect on program execution. It is possible, for example, to insert a higher sequence number between two lines.

```
N7   G00 G40 Z-20 M03 *
N8   X-12 Y+60 *
N9   G01 G42 X+20 Y+60 F40 *
N10  G 26 R5 F20 *
N11  X+50 Y+20 F40 *
N12  I-10 J+80 *
N13  G03 X+70 Y+51.715 *
```

Block

Each block in a program corresponds to one work step, for example:
N20 G01 G40 X+20 Y+30 Z+50 F1000 M03 *

Word

Each block is composed of words (e.g. X+20).

Address values

A word is composed of an address letter, e.g. X, and a value, e.g. +20.

The abbreviations in the above block have the following meanings:

N	= line number	X, Y, Z	= coordinates
G01	= linear interpolation, Cartesian	F	= feed rate
G40	= no tool radius compensation	M	= miscellaneous functions

Block format

Positioning blocks can contain:

- 8 G functions from various groups and also G90, G91 in front of each coordinate
- 3 coordinates and also 2 circle centers or pole coordinates
- 1 feed rate F
- 1 M function
- 1 spindle speed S
- 1 tool number

Fixed cycles can contain:

- Cycle parameter P (all files for the cycle definition)
- 1 M function
- 1 spindle speed S
- 1 tool number
- 1 positioning block (see above)
- 1 feed rate F
- Cycle call

Note

It is possible to combine fixed cycles with a positioning block, M-functions, spindle speed etc. (see example at right: long block format).

The short format, however, makes the program easier to read. This is especially important for fixed cycles.

Example: long format (not recommended)

```
:
N110  G75 P01+2 P02-20 P03-30
      P04 100 P05 X+50 P06 Y+20
      P07 200 T1 G17 S1000 G01
      X+40 Y+30 F250 M03 G79 *
```

Example: short format (recommended)

```
:
N110  T1 G17 S1000 *
N120  G01 X+40 Y+30 F250 M03 *
N130  G75 P01+2 P02-20 P03-30
      P04 100 P05 X+50 P06 Y+20
      P07 200 *
N40   G79 *
```

Programming in ISO

Editing functions



Editing

The term **editing** means entering, changing, supplementing and checking programs.

The editing functions are helpful in selecting and changing program blocks and words, and they become effective at the touch of a key.

Selecting a block



The current block stands between two horizontal lines.

A specific block is selected with "GOTO □".

Initiate the dialog



GOTO: NUMBER =





Key in and confirm the block number.

Paging through the program



Vertical cursor keys:

Select the next lower or next higher block number.

Hold down a vertical cursor key to continuously run through the program lines.

Inserting a block

You can insert new blocks anywhere in existing programs. Just call the block which is to precede the new block. The block numbers of the subsequent blocks are automatically increased.

If the program storage capacity is exceeded, this is reported at dialog initiation with the error message: = PROGRAM MEMORY EXCEEDED =.

This error message also appears if program end (PGM END block) is selected. You should then select a lower block number.

Editing words



Horizontal cursor keys:

The highlighted field is moved within the current block and can be placed on the program word to be changed.

One word in the current program block is to be changed:





Move the highlighted field to the word to be changed.

The dialog query appears for the highlighted word, e.g.

COORDINATES ?





Change the value.

To change another word:





Move the highlighted field to the word to be changed.

If all corrections have been made:





Conclude the block (or move the highlighted field to the right or left off the screen).

Programming in ISO

Editing functions



Searching for certain addresses

You can use the vertical cursor keys to search for blocks containing a certain address in the program.

Use the horizontal cursor keys to place the highlighted field on a word having the search address, and then page in the program with the vertical cursor keys:

only those blocks having the desired address are displayed.

Example

All blocks with the address M are to be displayed:	  	Select one block with the M.
	 	Place the highlighted field on a word with M.
MISCELLANEOUS FUNCTION M ?	  	Call blocks with the desired address M.

Programming in ISO

Clearing/deleting functions



Clear program



The dialog for clearing a program is initiated with the **CL PGM** key.

Initiate the dialog



ERASE = ENT/END = NOENT

Program is to be cleared:

↓

↑

 or

GOTO

 select a program number.

ENT

 Erase the program.

Program is not to be cleared:

NO ENT

Delete block



The current block (in a program) is deleted with **DEL** .

The block to be deleted is selected with **GOTO**  or a cursor key.

Program blocks can only be deleted in the PROGRAMMING AND EDITING operating mode.

After deletion, the block with the next lower sequence number appears in the current program line.

The following sequence numbers are corrected automatically.

The current program block is to be deleted:

DEL

 Delete block.

Delete program section

To delete program sections, call the last block of the program section.

Then continue pressing **DEL**  until all blocks in the definition or program section are deleted.

Clear entry, error message



You can clear numerical inputs with the “CE” key. A zero appears in the highlighted field after pressing the “CE” key.

Non-blinking **error messages can also be cleared with the “CE” key.**



An entered value and the address are completely cleared with “NO ENT”.



Program Selection

Opening a program

Selecting an existing program



You open a program and select a stored program by first pressing the “PGM NR” key (program number).

A table with the HEIDENHAIN dialog programs and ISO programs stored in the TNC appears on the screen. The program number last selected is highlighted. The program length in characters is given after the program number. ISO programs are designated by “ISO” after the program number.

You can select the desired program either

- via the cursor keys
- or
- by entering its number.

If the selected program number does not yet exist, a new program is opened.

PROGRAM SELECTION			
PROGRAM NUMBER =			
1	360		
10001	756		
10002	1440		
111	1548		
11111	44	ISO	
13	450		
14	450		
2	900		

ACTL.	X +	49,258	Y + 23,254
	Z +	15,321	C + 84,000
			0 MS/9

Opening a program



Depending on the selected program type, HEIDENHAIN dialog programs or ISO programs can be opened (see index A “General Information, MOD Functions”).

Initiate the dialog



PROGRAM SELECTION

PROGRAM NUMBER =

MM = G71 / INCH = G70

% 231 G

ENT

Enter the program number (maximum 8 characters). Confirm entry.

7

1

for dimensions in mm, or

7

0

for dimensions in inches.

END

Example display

% 231 G71 *
N9999 % 231 G71 *

Selecting an existing program



All existing programs (HEIDENHAIN format and ISO) can be edited, tested, displayed graphically and executed, regardless of the selected type of programming.

Initiate the dialog



PROGRAM SELECTION

PROGRAM NUMBER =

or

↓

↑

ENT

Place the highlighted field on the desired program number.

ENT

Enter the program number.

Example display

0 % 231 G71 *
1 N10 G30 G17 X+0 Y+0 Z-40 *
2 N20 G31 G90 X+100 Y+100 Z+0 *

Program Selection

Erase/edit protection: G50



Edit protection G50

After creating a program, you can designate it as erase- and edit-protected. Protected programs can be executed and viewed, but not changed. A protected program can only be erased or changed if the erase/edit protection is removed beforehand. This is done by selecting the program and entering the code number **86357**.

Activating edit protection

Initiate the dialog

PROGRAM NUMBER =

Enter the number of the program to be protected, confirm entry.

% 7210 G71 G

Press the key until the dialog query "PGM protection" appears.

PGM PROTECTION ?

Protect the program.

Confirm block.

% 7210 G71 G50 *

Erase/edit protection is programmed. **G50** appears at the end of the line.

Removing edit protection

Initiate the dialog

PROGRAM NUMBER =

Enter the number of the program whose edit protection is to be removed.

% 7210 G71 G50 *

Select the auxiliary operating mode.

VACANT MEMORY: 148330 BYTE

Select the MOD function "Code number".

CODE NUMBER =

Enter code number **86357**.

% 7210 G71 *

Erase/edit protection is removed. "G50" is deleted.

Code number 86357

Program Selection

Defining the workpiece blank: G30/G31



Test graphics

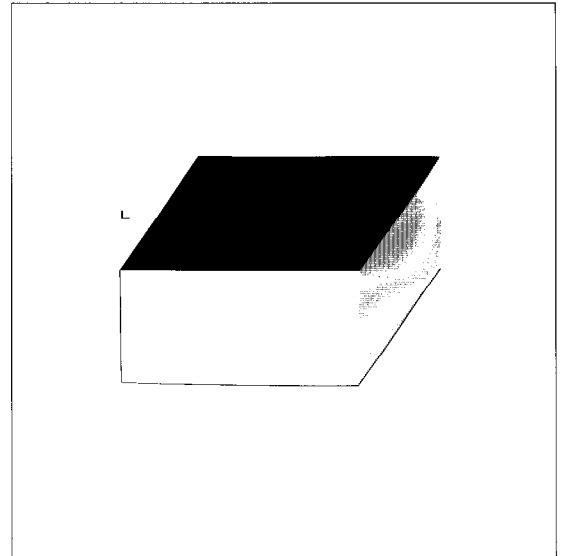
A blank form definition must be programmed before the machining program can be simulated graphically.

Blank

For the graphic displays, the blank dimensions of the workpiece must be entered at the start of program via G30/G31.

The blank form must always be programmed as a cuboid, aligned with the machine axes.

Maximum dimensions:
14000 x 14000 x 14000 mm.

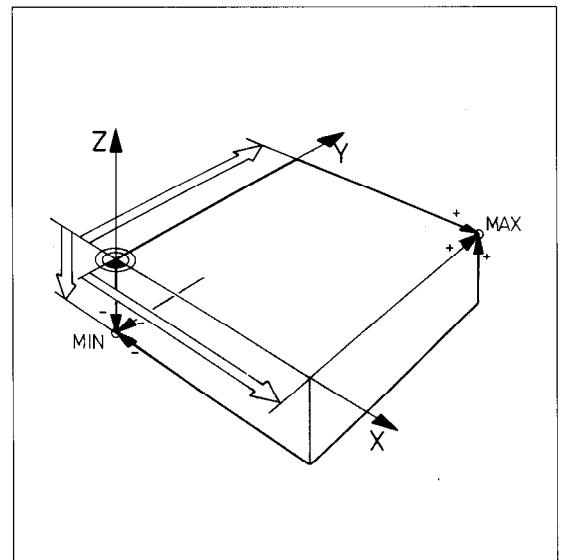


Minimum point Maximum point

The cuboid is defined with the minimum point (**MIN**) and maximum point (**MAX**) (points with "minimum" and "maximum" coordinates).

MIN can only be entered in absolute dimensions;
MAX may also be incremental.

The blank data are stored in the associated machining program and are available after program call.



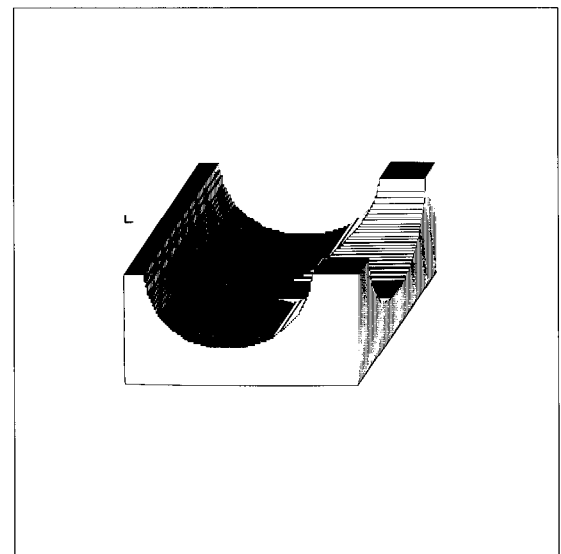
Graphic display

Machining can be simulated in the three main axes – with a fixed tool axis.

Tool form

The graphic simulation depicts the results of machining with a cylindrical tool.

The graphic must be interpreted accordingly when using form tools.



Program Selection

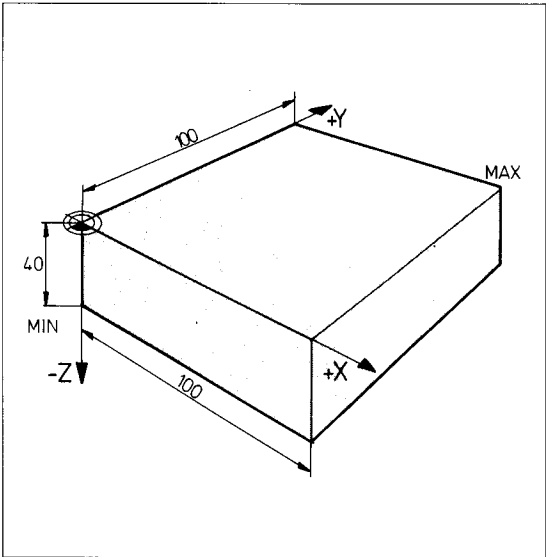
Defining the workpiece blank: G30/G31



Example The blank form is aligned with the main axes.

The **MIN** point has the coordinates X0, Y0 and Z−40.

The **MAX** point has the coordinates X100, Y100 and Z0.



Note To define a blank, a program must be selected in the "Programming and editing" operating mode.

Entering the cuboid corner points MIN

▶ **G 3 0** Blank form definition for MIN point.
▶ **G 1 7** Tool axis Z.
▶ **X 0** X coordinate.
▶ **Y 0** Y coordinate.
▶ **Z +/− 4 0** Z coordinate.
▶ **END** Conclude block.

MAX

▶ **G 3 1** Blank form definition for MAX point.
▶ **G 9 0** Absolute dimensions.
▶ **X 1 0 0** X coordinate.
▶ **Y 1 0 0** Y coordinate.
▶ **Z 0** Z coordinate.
▶ **END** Conclude block.

Example display N10 G30 G17 X+0 Y+0 Z−15 *
N20 G31 G90 X+100 Y+100 Z+0 *

Error messages **BLK FORM DEFINITION INCORRECT**
The MIN and MAX points are incorrectly defined, or the machining program contains more than one blank definition, or the side proportions differ too greatly.

PGM SECTION CANNOT BE SHOWN
Wrong spindle axis is programmed.

Tool Definition

Tool definition within the part program



Tool definition



The control requires the tool length and tool radius to enable it to compute the tool path from the given work contour.

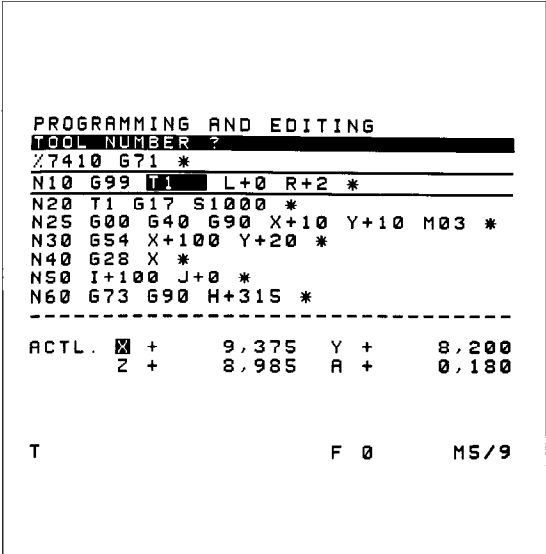
These data are programmed in the tool definition.

Whether the tools are defined decentralized in the appropriate part program or in a central tool file (program 0) is determined by a machine parameter.

Tool number

Compensation values always refer to a certain tool which is designated by a number.

Valid tool numbers:
with automatic tool change or in program 0:
1 to 99
without automatic tool change or in the machining program: 1 to 254.



Tool definition in the part program

If tools required in a program are defined in that program, a program printout will include the specifications of the tool dimensions.

Input

Initiate the dialog



TOOL NUMBER ?

☐ Enter the tool number.

The tool number 0 cannot be programmed under G99.

Tool 0 is internally defined with L = 0 and R = 0.

TOOL LENGTH L ?

☐ Enter the tool length or the difference to the zero tool.

TOOL RADIUS R ?

☐ Enter the tool radius.

Conclude the block.

Tool Definition

Tool definition in program 0



Central tool file

If the central tool file (program 0) is activated by machine parameters, the tools must always be defined there.

They then only have to be called in any program.

The central tool file is programmed, changed, output and read in the "Programming and editing" operating mode.

Every tool is entered with the tool number, length, radius and pocket number. Tool 0 must be defined with L = 0 and R = 0.

PROGRAMMING AND EDITING					
T2	L+5,3	R+6			
T3	L+12,45	R+7,75			
T4	L+25,21	R+3,5			
T5	L+52,52	R+2,5			
T6	L+85	R+2			
T7	L+32,71	R+8			
T8	L+147,1	R+13			
T9	L+0	R+15,5			

ACTL.	X + 49,258	Y + 23,254			
	Z + 15,321	C + 84,000			
				F 0	M5/9

Example

Tool 3 is to be defined with L = 5, R = 7:

Initiate the dialog



PROGRAM NUMBER ?

Select the central tool file.

BEGIN TOOL MM

Select the tool.

T3 L0 R0

Enter the length.
Enter the radius.

Tool changer with flexible pocket coding

On machines with a tool magazine and flexible pocket coding, the tools can be returned to a different magazine pocket than they were taken from.

The control memorizes which tool number is stored in which pocket.

G99 functions like a tool pre-selection here, i.e. the tool search is programmed with G99. In this case, only the query for the tool number appears.

Oversize tools

Oversize tools occupying three pockets are to be designated as "special tools". A special tool is always returned to the same pocket.

Program by placing the highlighted field on the dialog query

SPECIAL TOOL ?

and respond with the "ENTER" key.

The preceeding and succeeding pocket numbers should be deleted by positioning the highlighted field and pressing the "NO ENT" key. A * is displayed in place of the erased pocket number.

"S" for special tool and "P" for pocket number only appear if this function was selected via machine parameters.

P0 (spindle) or another pocket must be vacant in the magazine.



Tool length L

The tool length is compensated with a single adjustment of the spindle axis by the length compensation.

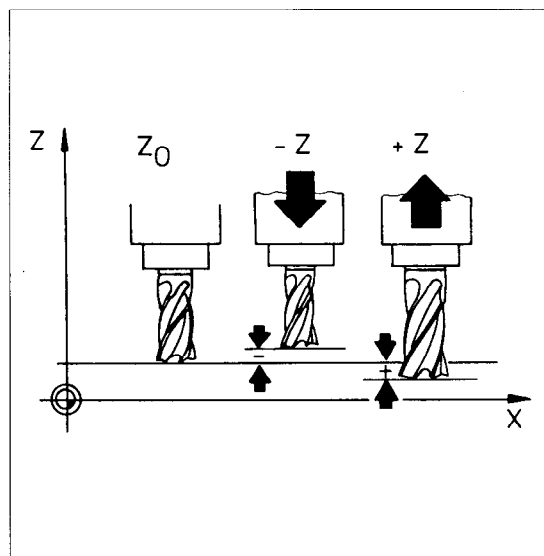
Compensation becomes effective after tool call and subsequent movement of the tool axis.

Zero tool

Compensation ends after a tool is called or with T_0 (T_0 is called the zero tool and has a length of 0).

The correct compensation value for the tool length can be determined on a tool pre-setter or on the machine.

If the compensation value is to be determined on the machine, then you must first enter the work-piece datum.



Length differences

When the compensation values are determined on the machine, the zero tool serves as a reference.

The length differences $-Z$ or $+Z$ of the other clamped tools to this zero tool are programmed as tool length compensations.

If a tool is **shorter** than the zero tool, the difference is entered as a **negative** tool length compensation. If a tool is **longer** than the zero tool, the difference is entered as a **positive** tool length compensation.

Preset tools

If a tool pre-setter is used, all tool lengths are already known. The effective compensation values correspond to the tool length and are entered with the correct signs according to a list.



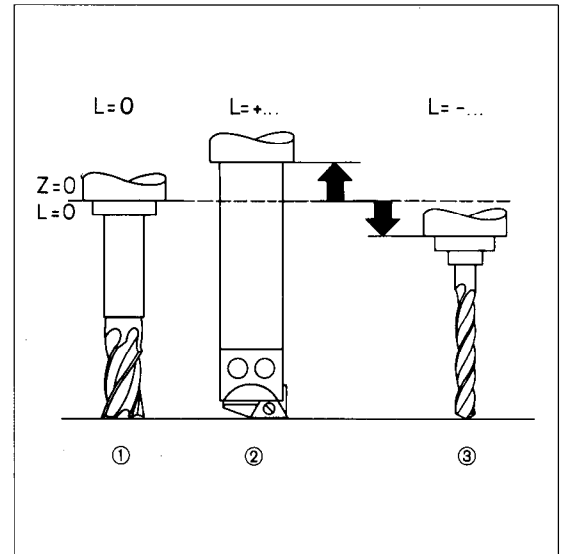
Tool Definition

Transferring tool length



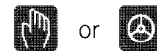
Tool lengths can be easily and quickly entered with the "teach in" function.

1. Move the zero tool T_0 to the work surface and set the spindle axis to zero.
2. After exchanging, move the tools T_1 or T_2 to the work surface.
3. Transfer each display value in this position to the tool length definition. This gives you the length compensation to the zero tool.



Input

Operating mode



or



Touch the surface with the zero tool.

Initiate the dialog



Spindle axis, e.g. Z.

DATUM SET



Reset to zero.



Also touch the surface with the new tools T_1 or T_2 .

Operating mode



Either

1. call a tool definition in a program and initiate the dialog "TOOL LENGTH L ?",

or

2. select a tool in the central tool file and initiate the dialog "TOOL LENGTH L ?".

TOOL LENGTH L ?



Select the spindle axis to transfer the tool length.



Transfer the length compensation to memory.

TOOL RADIUS R ?



Enter the radius.



Tool radius R

The tool radius is entered as a positive number (exception: radius compensation when programming the cutter center path).

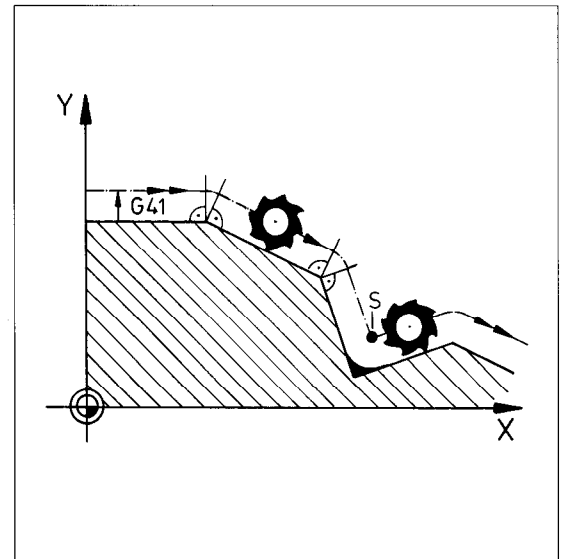
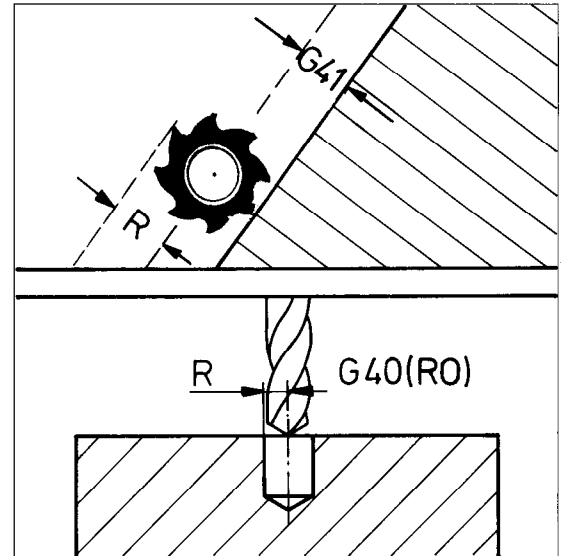
A tool radius must always be programmed before a machining program can be checked with test graphics.

Tool radius compensation

Drilling work is programmed without radius compensation (G40), while milling jobs are usually programmed with radius compensation (G41/G42).

Compensation is effective after a tool call, programming with G41 or G42 in a positioning block (G01, G02 etc.), or a movement in the active interpolation plane. Compensation ends with a positioning block which contains G40.

If the tool travels with path compensation, i.e. the tool center path is offset by the programmed tool radius, the tool follows a path parallel to the contour at the distance of the tool radius. The programmed feed rate applies to the center path.



Outside corners

The control inserts a transition curve for the center path of the tool at outside corners, so the tool rolls around the corner.

In most cases, the tool is thus guided at a constant path speed around the outside corner.

Automatic deceleration at corners

If the programmed feed rate is too high for the transition curve, the path speed is reduced (which produces a more precise corner). The limit value is permanently programmed in the control (machine parameter).

Inside corners

The control automatically determines the intersection S of the two cutter paths parallel to the contour (equidistant) at inside corners.

This prevents back-cutting in the contour; the work is not damaged. The control thus shortens traversing distances according to the tool radius in use.

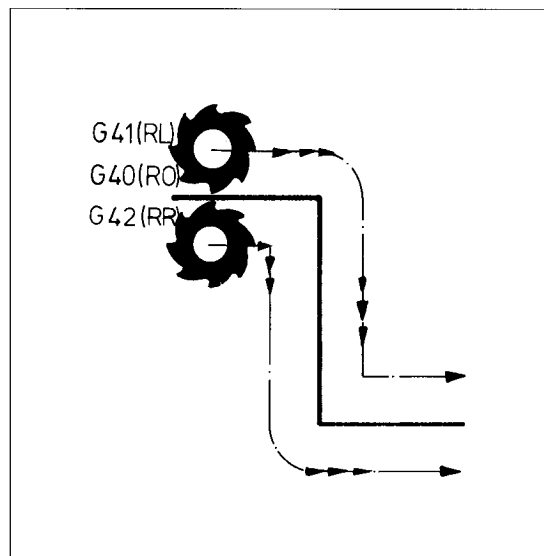
The radius of the tool must always be chosen so that every contour element – even when shortened – can be machined.

Cutter Path Compensation

Entering the radius compensation



To automatically compensate for the tool radius – as entered in the TOOL DEF blocks – the control must be informed whether the tool travels to the left of, to the right of, or directly on the programmed contour.



G40 (R0)

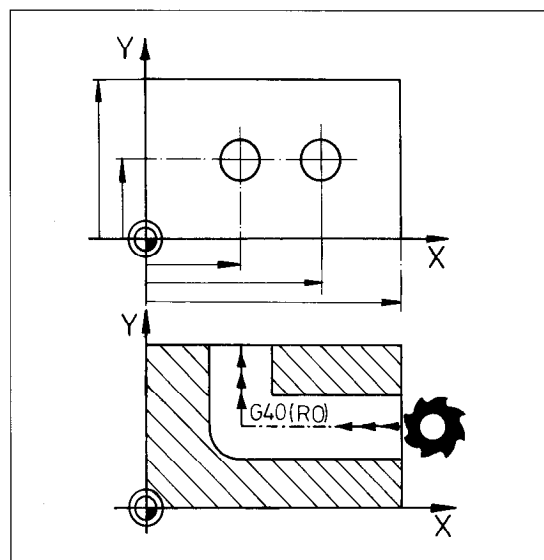


If the tool is to travel **on** the programmed contour, no radius compensation should be programmed in the positioning block. The modal function G40 (R0) must therefore be programmed in the same or in a previous block.

Programming radius compensation

The radius compensation is entered in positioning blocks (G01, G02 etc.) via the functions **G41** (RL) and **G42** (RR).

"Left" or "right" should be understood as looking in the direction of movement.



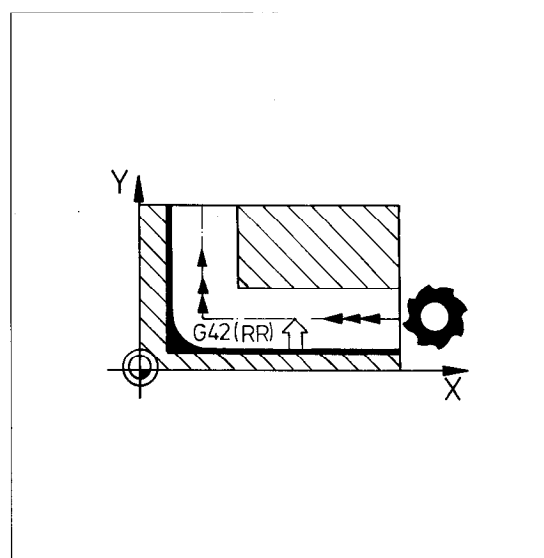
G41 (RL)

If the tool is to travel at the distance of the radius to the left of the programmed contour, enter the function G41 (RL).

G42 (RR)

If the tool is to travel at the distance of the radius to the right of the programmed contour, enter the function G42 (RR).

The functions G40, G41 and G42 are modal, which means that they remain effective for all following blocks until changed. If you wish to keep the radius compensation of the previous block, no entry is necessary.



Cutter Path Compensation

Working with radius compensation



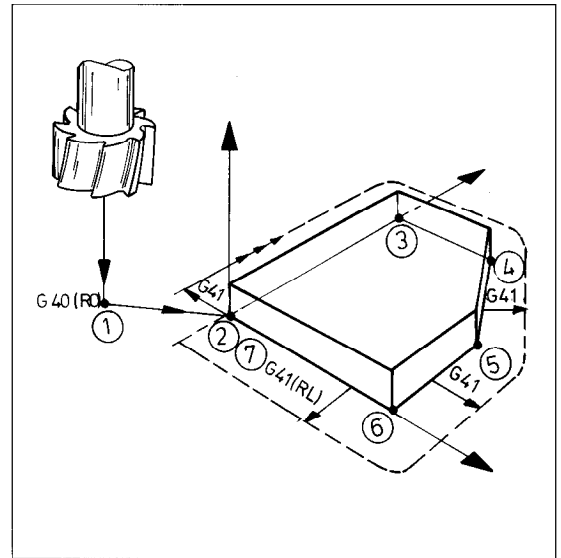
Starting point G40 (R0)

Change the tool and call the compensation values with "TOOL CALL".

Traverse rapidly to the starting point ①.

At the same time lower Z to the working depth (if danger of collision, first traverse in X/Y, then separately in Z!). This compensates for the tool length.

The radius compensation still remains switched off with G40.



1st contour point G41 (RL) G42 (RR)

Traverse to contour point ② with radius compensation G41 (RL) or G42 (RR) at reduced feed rate.

Machining around the contour

Program the following contour points to ⑦ at milling feed rate.

Since the radius compensation remains unchanged, there is no further need to enter G41 or G42 until point ⑦.

Last contour point G41/G42

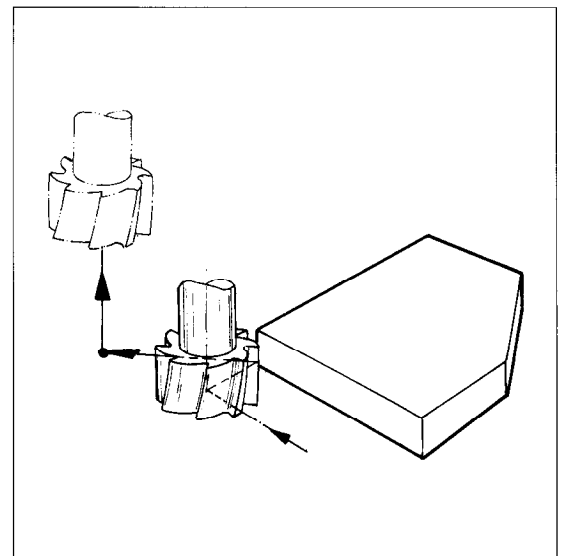
After a complete circulation, the last contour point ⑦ is identical to the first contour point ② and is still radius compensated.

End point G40

The end point (outside the contour) must be programmed without compensation with G40 to complete machining.

To prevent collisions, retract only in the machining plane to cancel the radius compensation.

Then back-off the tool axis separately.



Cutter Path Compensation

Radius compensation G43/G44



G43 (R+) G44 (R-)

By entering G43 (R+) or G44 (R-) you can lengthen or shorten a **paraxial** movement (i.e. movement in only one axis) by the length of the tool radius.

This simplifies:

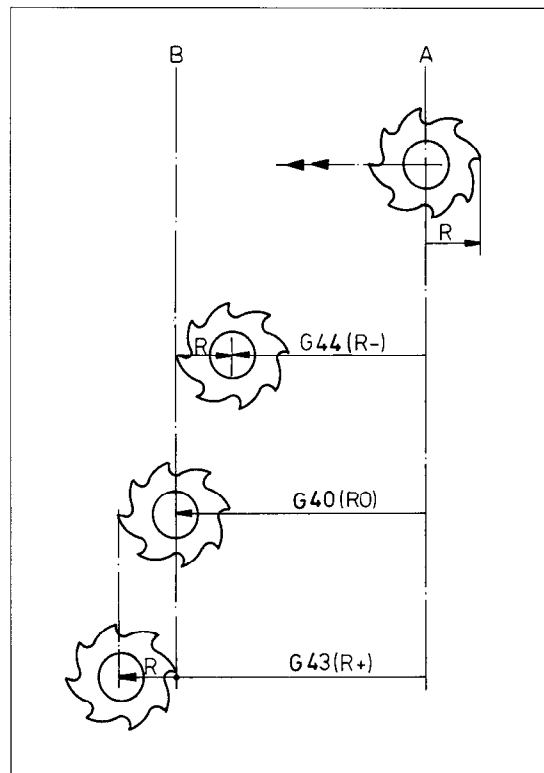
- Positioning with manual data input
- Paraxial positioning
- Pre-positioning for the "slot" cycle.

Effect G43/G44

This radius compensation has the following effect:

- The displacement is **lengthened** by the tool radius: display **G43**.
- The tool traverses **to** the programmed nominal position: display **G40**.
- The displacement is **shortened** by the tool radius: display **G44**.

G43/G44 do not affect the spindle axis.



Example

The tool is to traverse from initial position $X = 0$ to $X = (46 + \text{tool radius})$.

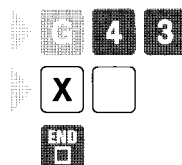
Application example:

Pre-positioning for the "Slot" cycle.

Input



Paraxial positioning.



Paraxial compensation, e.g. lengthening (R+).
Nominal position value, e.g. $X+46$.
Conclude block.

Display

G07 G43 X+46 *

Mixing

G01 and

Uncompensated blocks (e.g. G01 G40 X+20) and paraxial blocks (e.g. G40 X+20 or G43 X+20) can be mixed in a part program.

Paraxial compensated positioning blocks (G43/G44) and radius compensated positioning blocks (G41/G42) are not to be entered in succession!

Correct:

```
G01 G40 X+15 Y+20 *
    G40 Y+50 *
    G43 X+40 *
    G40 Y+70 *
```

Incorrect:

```
G01 G42 X+15 Y+20 *
    G43 Y+50 *
    G42 X+50 Y+57 *
```

Tools

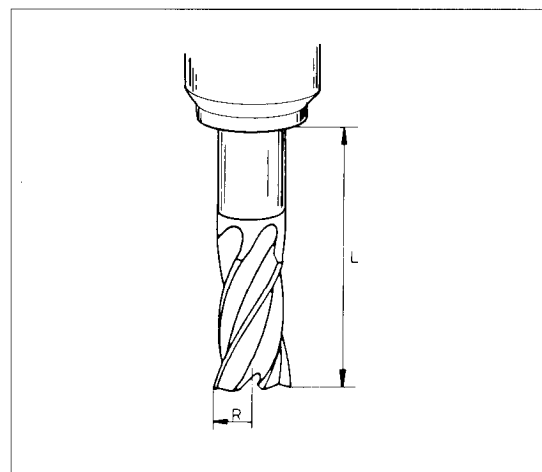
Tool call



Tool call



With the "T" key a new tool and the associated compensation values for length and radius are called up.



Spindle axis

In addition to the tool number, the control also needs to know the spindle axis to carry out length compensation in the correct axis or radius compensation in the correct plane.

Compensation effect

The spindle axis also defines the plane (e.g. XY) for circular movements: It is identical to the "radius compensation" plane. This is also the plane for "coordinate rotation" and "mirror image".

Spindle axis	Length compensation	Radius compensation
Z (G17)	Z	XY
Y (G18)	Y	ZX
X (G19)	X	YZ

Spindle speed

The spindle speed is entered directly after the spindle axis.
Input range of the control: 0 to 99999 rpm.
If the speed exceeds the valid range for the machine, the following error message appears at program run
WRONG RPM

Activating compensation

A tool call activates length compensation.
It first becomes effective when the next tool axis movement is programmed.
It can be seen as a single movement in the tool axis.
Radius compensation first becomes effective when the compensation direction G41 or G42 is programmed in a positioning block.

Ending compensation

A tool call block (T-block) ends the "old" tool length and tool radius compensation and calls the new compensation values.
Example: T12 G17 S300 *
Tool radius compensation is also ended by programming G40 in the positioning block.
If only the spindle speed is entered with a tool call block, the compensations remain valid.
Example: T12 S500 *

Tool call

Initiate the dialog



TOOL NUMBER ?			Enter the tool number.
			Enter the spindle axis, e.g. G17.
			Enter the spindle speed in rpm.
			Conclude the block.

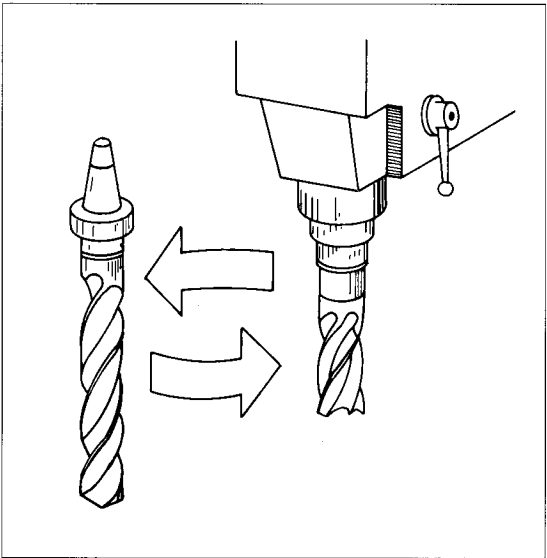
Tools

Tool change

Tool change position

To change the tool, the main spindle must be stopped and the tool retracted in the spindle axis.

We recommend programming an additional block in which the axes of the machining plane are likewise backed-off.



Workpiece-related change position

The tool moves to a **workpiece-related** position if no additional measures are taken.

Example: `G00 Z+100 M06 *`

The tool is driven 100 mm over the work surface if the tool length is 0 or T0 was programmed.



T0 reduces the distance to the workpiece (danger of collision!) if a positive length compensation was effective prior to TOOL CALL 0.

Machine-related change position

You can use M91, M92 or a PLC positioning to traverse to a **machine-related** tool change position.

Example: `G00 Z+100 M92 *`

(see "Predetermined M Functions, Machine-referenced coordinates M91/M92").

Manual tool change

The program must be stopped for a manual tool change. Therefore, enter a program STOP before the tool call (T-block). M6 has this stop effect when the control is set accordingly via machine parameters. The program is then stopped until the machine START button is pressed.

The program STOP can only be omitted when a tool call is programmed solely to change the spindle speed.

Automatic tool change

The tool is changed at a defined change position. The control must therefore move the tool to a machine-referenced change position. The program run is not interrupted.

```
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T1 L+0 R+5 *
N40 G99 T2 L-2.4 R+3 *
N50 T0 G17 *
N60 G00 G40 G90 Z+200 M06 *
N70 T1 S1000 *
N80 X+25 Y+30 *
N90 Z+2 M03 *
```

Feed Rate F/Spindle Speed S/ Miscellaneous Functions M



Feed rate



The feed rate F, i.e. the traversing speed of the tool in its path, is programmed in positioning blocks in mm/min or 0.1 inch/min. The current feed rate is shown in the status display on the lower right of the screen.

Feed rate override

The feed rate can be varied within a range of 0% to 150% with the feed rate override on the control operating panel. The effective range of the potentiometer for tapping is limited by machine parameters!

Rapid traverse

The maximum input value (rapid traverse) on the control for positioning is:

- 29998 mm/min or
- 11800/10 inch/min.

The maximum operating speeds are set for each axis.

G00 or the max. input is programmed for rapid traverse.

The control automatically limits rapid traverse to the permissible values.



If the F display is highlighted and the axes do not move, this means the feed rate was not enabled at the control interface. In this case, you must contact your machine manufacturer.

Spindle speed



The spindle speeds are set through a tool call (T-block).

Spindle override

On machines with continuous spindle drive, the speed can be varied from 0% to 150% using the spindle override.

Spindle override is disabled during tapping.

Miscellaneous functions



Miscellaneous functions can be programmed to regulate certain machine functions (e.g. spindle "on"), to control program run and to influence tool movements. The miscellaneous functions are comprised of the address M and a code number according to ISO 6983. All of the M functions from M00 to M99 can be used.

Certain M functions become effective at the start of block (e.g. M03: spindle "on" clockwise), i.e. before movement, and others become effective at the end of block (e.g. M05: spindle "stop").

Only a certain number of these M functions are effective on any given machine.

Some machines may employ additional, non-standard M functions not defined by the control.

M functions are normally programmed in positioning blocks (G01, G02 etc.).

However, M functions can also be programmed without positioning.

Path Movements

Input



Contour elements

The coordinates which you enter must describe the shape of the workpiece, not the path of the tool center. The control compensates for the tool radius and computes the centerline of the tool path required to machine the programmed contour.

You program as if the tool is always moving and the workpiece is always stationary, regardless of the actual design of you machine tool. The programmable contours are composed of the contour elements **straight line** and **circle**.

Generating the workpiece contour

To be able to compute the workpiece contour, the control must be given the individual contour elements. Since each program block specifies the next step, the following information is required.

- straight line or circle
- the coordinates of each end point
- additional information such as circle center, contour radius etc.

The following is an example of positioning block input for a straight line.

Input		1	Selection of type of movement, e.g. linear Cartesian.
		0	No radius compensation (R0) or
		1	radius compensation left (RL) or
		2	radius compensation right (RR).
First coordinate		0	Absolute or
		1	incremental.
Next coordinate			Coordinate and value.
			Feed rate.
			M function.
			Conclude block.

Example

N20 G01 G40 G90 X+20 Z-10 G91 Y+30 F100 M03 *
Linear, Cartesian, no radius compensation (G40), absolute to X+20, Z-10, incremental to Y+30 with feed rate 100 and spindle on clockwise.

Abbreviated input

G functions, for example G01, G40, G90, feed rates and some M functions are modal, that is they remain active until they are cancelled or replaced with another function of the same type.

Example

N20 G01 G40 G90 X+20 F100 M03 *
N30 Y+30 *

Path Movements

Overview of path functions



Straight lines

Circles

Function	Input	
	in Cartesian coordinates	in polar coordinates
Straight line movement in rapid traverse	G00	G10
Straight line movement at programmed feed rate	G01	G11
Chamfer with length R A chamfer is inserted between two straight lines	G24	
Circle center; also pole for programming polar coordinates I, J, K do not generate movement	I, J, K	
Circular movement in clockwise direction (CW)	G02	G12
Circular movement in counterclockwise direction (CCW)	G03	G13
The circular path can be programmed: • circle center I, J, K and end point, or • circle radius and end point		
Circular movement without indication of direction of rotation. Only the radius and end point of the circular path need to be programmed. The direction of rotation results from the circular movement G02/G12 or G03/G13 which was last programmed	G05	G15
Circular movement with tangential transition. An arc is attached to the preceding contour element with a tangential transition. Only the end point of the arc needs to be programmed.	G06	G16
Rounding corners with radius R. An arc with tangential transitions is inserted between two contour elements.	G25	

Multi-axis movements

A maximum of 3 axes can be programmed for straight lines, and a maximum of 2 axes for circles.

Path Movements

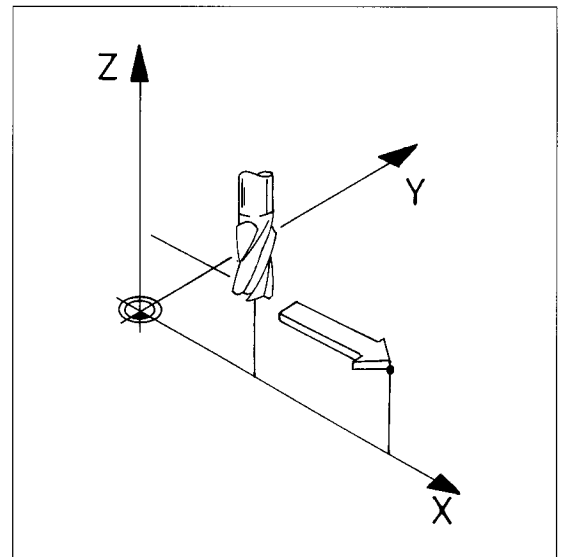
1D/2D/3D movements



Movements are referred to – depending on the number of simultaneously traversed axes – as 1D, 2D or 3D movements (D for “dimension”).

Paraxial traverse: 1D movements

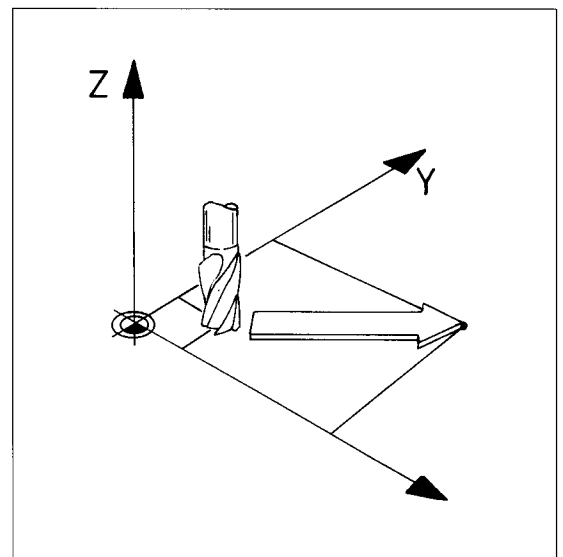
If the tool is moved relative to the work on a straight line parallel to a machine axis, this is called paraxial positioning or machining.



2D movements

Movement in a main plane (XY, YZ, ZX) is called 2D movement.

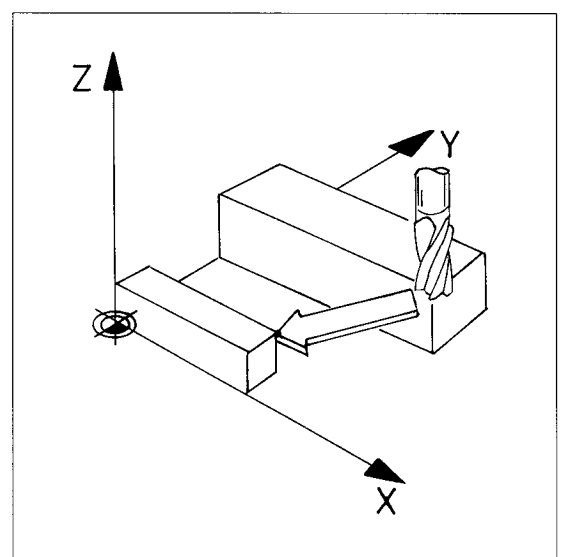
Straight lines and circles can be generated in the main planes with 2D movements.



3D movements

If the tool is moved relative to the workpiece on a straight line with simultaneous movement of all three machine axes, it is called a 3D straight line.

3D movements are required to generate oblique planes and bodies.



Linear Movement, Cartesian

Positioning in rapid traverse: G00



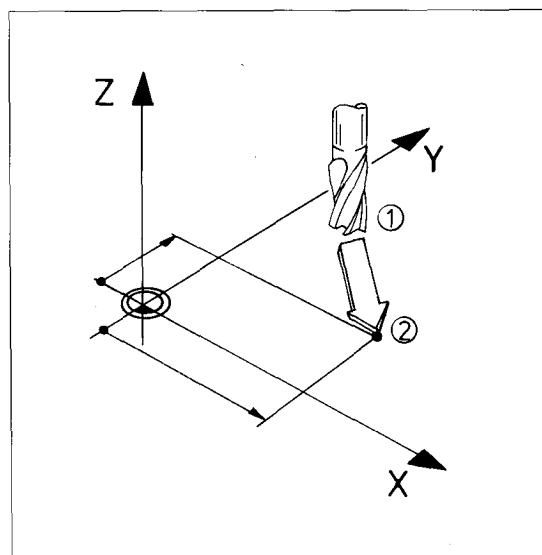
Positioning

The tool is at the starting point ① and must travel on a straight line to target point ②.

You always program the target point ② (nominal position) of straight lines.

Position ② can be entered in Cartesian or polar coordinates.

The first position in a program must always be entered as an absolute value. The following positions can also be incremental values.



Example tool definition/ call

G99 T1 L+10 R5 *

T1 G17 S200 *

Tool 1 has length 10 mm and radius 5 mm.

Tool 1 is called in the spindle axis Z.
Spindle speed is 200 rpm.

Positioning block: complete input (main block)

G 0
G 40 G 90
X 50 Y 30 Z 0
M 3

Rapid traverse.

No radius compensation, absolute dimensions

Z is moved with tool length compensation

Spindle clockwise.

G00 G40 G90 X+50 Y+30 Z+0 M3 *

Re-entry at tool calls is especially easy if you enter a main block (= complete positioning block) after a tool call.

The G function for positioning in rapid traverse (G00 or G10) is modal. Beware of collision during tool downfeed.

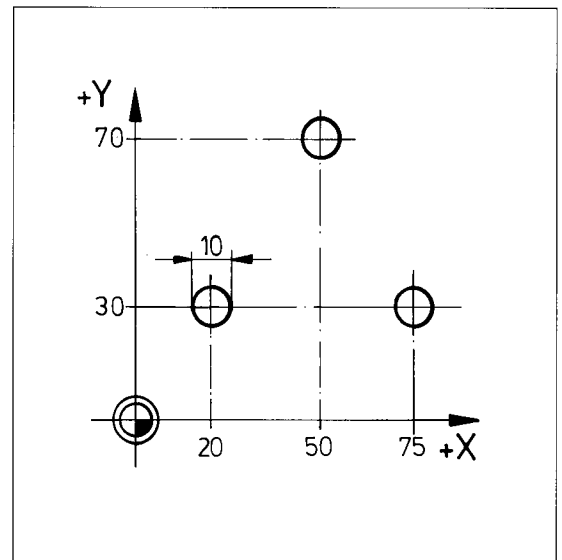
Linear Movement, Cartesian Drilling: G01



Absolute Cartesian coordinates

1 20 30 2

G01 X+20 Y+30 Z+2 *



Incremental Cartesian dimensions

1 91 20

Only incremental entry.

G01 G91 X+20 *

Mixed entries

1 91 20 90 30

The position for X is entered in incremental dimensions, for Y in absolute dimensions.

G01 G91 X+20 G90 Y+30 *

Example drilling

The following is an example of a program for drilling without cycles.

Program

%10 G71 *

N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T1 L+0 R+5 *
N40 T1 G17 S2400 *
N50 G00 G90 Z+200 M6 *

N60 G40 X+20 Y+30 M3 *

N70 Z+2 *
N80 G01 Z-10 F80 *
N90 Z+2 F1000 *
N100 G00 X+50 Y+70
N110 G01 Z-10 F80 *
N120 Z+2 F1000 *
N130 G00 X+75 Y+30 *
N140 G01 Z-10 F80 *
N150 G00 Z+200 M2 *
N9999 %10 G71 *

Blank form definition (only if graphic workpiece simulation desired)

Tool definition

Tool call

Retract in Z,

tool change

Positioning to 1st hole in X/Y,
rapid traverse, switch on spindle

Pilot positioning in Z

Drilling at programmed feed rate

Retract in Z

Positioning to 2nd hole in X/Y

Drilling at programmed feed rate

Retract in Z

Positioning to 3rd hole in X/Y

Drilling at programmed feed rate

Retract in Z

End of program

Linear Movement, Cartesian

Chamfer: G24



Chamfer G24

A chamfer can be programmed for contour corners formed by the intersection of two straight lines. The angle between the two straight lines can be arbitrary.

Prerequisites

A chamfer is completely defined by the points ① ② ③ and the chamfer block. A positioning block containing both coordinates of the machining plane should be programmed before and after a chamfer block. The compensation G40/G41/G42 must be identical before and after the chamfer block. A contour cannot be started with a chamfer.

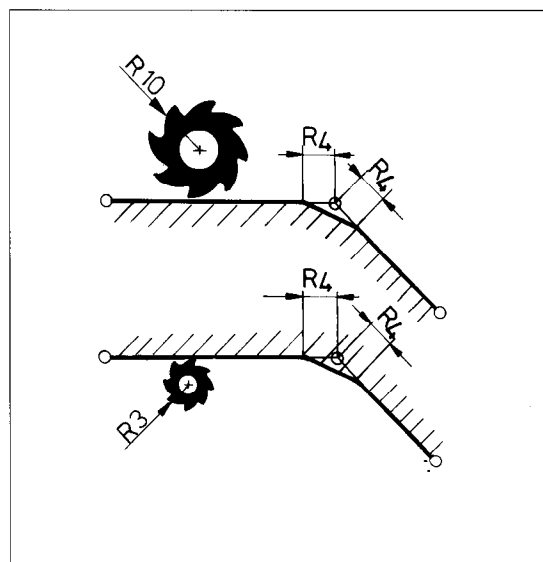
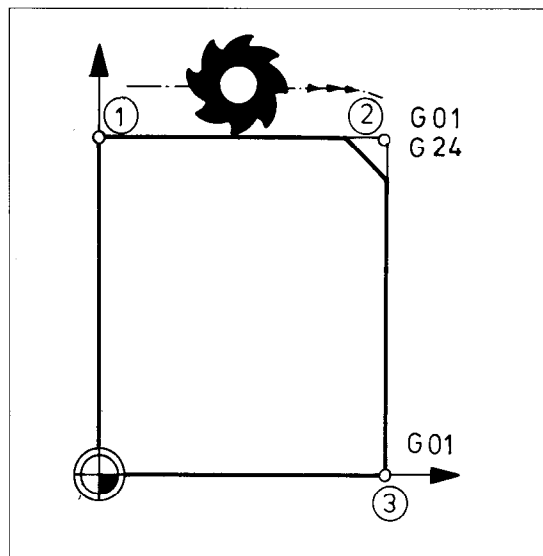
A chamfer can only be executed in the machining plane. The machining plane in the positioning block before and after the chamfer block must therefore be the same.

The chamfer length must not be too long or too short at inside corners: the chamfer must "fit between the contour elements" and also be machineable with the chosen tool.

The previously programmed feed rate remains effective for the chamfer.

Programming

Program a chamfer as a separate block. Only enter the chamfer length – **no coordinates**. The "corner point" itself is not traversed!



R = chamfer length

Entering the chamfer



Program block

G24 R4 *

Example

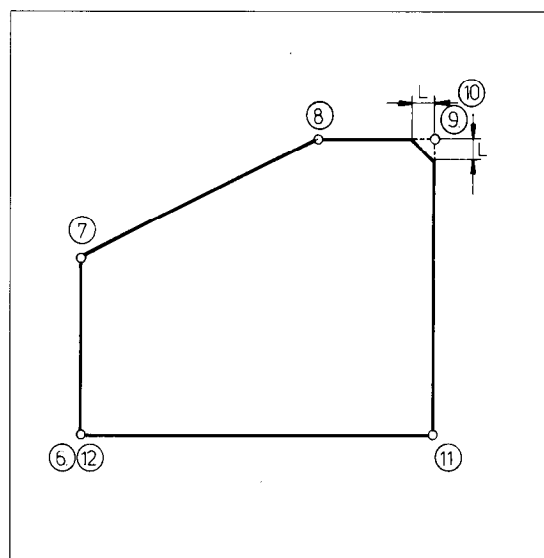
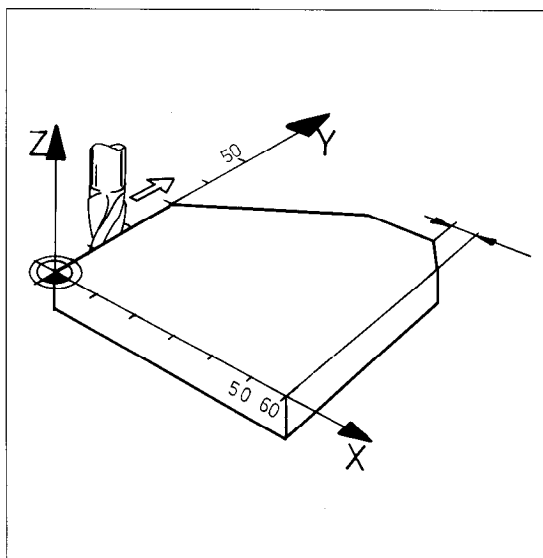
```
%11 G71 *
N10 G99 T1 L+0 R+10 *
N20 T1 G17 S200 *
N30 G01 G41 X+0 Y+50 F300 M03 *
N40 X+50 Y+50 *
N50 G24 R4 *
N60 X+50 Y+0 *
N9999 %11 G71 *
```

Position ① (see figure above)
Position ②
Chamfer
Position ③

Linear Movement/Cartesian Example



**Example:
milling
straight
lines**



The block numbers are shown in the figure to aid you in following the sequence.

Program

```
%12 G71 *
N3 G30 G17 X+0 Y+0 Z-40 *
N5 G31 G90 X+100 Y+100 Z+0 *
N10 G99 T1 L+0 R+5 *
N20 T1 G17 S500 *
N30 G00 G90 Z+200 M06 *
N40 G40 X-10 Y-20 M03 *
N50 G01 Z-20 F80 *
N60 G41 X+0 Y+0 F200 *
N70 Y+30 F400 *
N80 X+30 Y+50 *
N90 X+60 *
N100 G24 R5
N110 Y+0 *
N120 X+0 *
N130 G40 X-20 Y-10 *
N140 G00 Z+200 M02 *
N9999 %12 G71 *
```

Blank form definition (MIN point)
Blank form definition (MAX point)
Tool definition
Tool call
Tool change
Pilot position (tool is up)
Plunge at downfeed rate
Approach the contour, call radius compensation ⑥
Machine the contour ⑦
⑧
⑨
Chamfer block ⑩
○
○
Last block with radius compensation
Cancel radius compensation
Back-off Z

Linear Movement, Cartesian

Additional axes



Linear axes U, V, W

Linear interpolation can be performed simultaneously with a maximum of 3 axes – even when using additional axes.

For linear interpolation with an additional linear axis, this axis must be programmed with the corresponding coordinate **in every NC block**. This requirement holds even when the coordinate remains unchanged from one block to the next. If the additional axis is not specified, the control traverses the main axes of the machining plane again.

Example: linear interpolation with X and V,
tool axis Z.

```
N11 G01 G42 X+0 V+0 F100 *
N12 X+100 V+0 *
N13 X+150 V+70 *
```

Rotary axes A, B, C

If the additional axis is a rotary axis (A, B or C axis), the control registers the entered value in angular degrees.

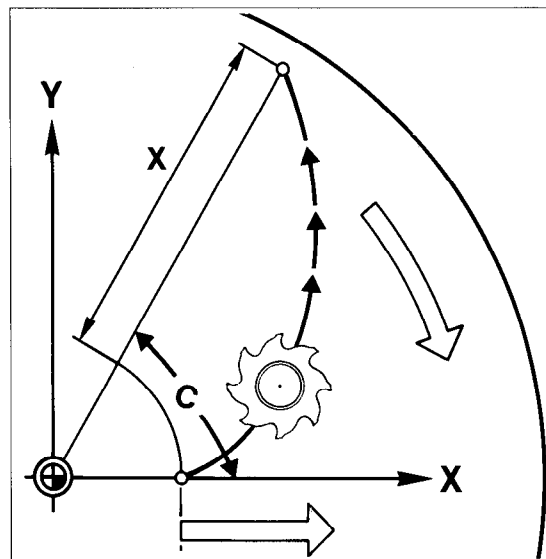
During linear interpolation with one linear and one rotary axis, the TNC interprets the programmed feed rate as the path feed rate. That is, the feed rate is based on the relative speed between the workpiece and the tool. Thus, for every point on the path, the control computes a feed rate for the linear axis F_L and a feed rate for the angular axis F_W :

$$F_L = \frac{F \cdot \Delta L}{\sqrt{(\Delta L)^2 + (\Delta W)^2}}$$

$$F_W = \frac{F \cdot \Delta W}{\sqrt{(\Delta L)^2 + (\Delta W)^2}}$$

where:

- F = programmed feed rate
- F_L = linear component of the feed rate (axis slides)
- F_W = angular component of the feed rate (rotary table)
- ΔL = linear axis displacement
- ΔW = angular axis displacement



M94 for rotary axes

The position display for rotary axes can be set via machine parameters for either:

- $\pm 360^\circ$ or
- $\pm \infty$ (i.e. $\pm$ max. display value).

If $\pm \infty$ is chosen as the measuring range, the position display for rotary axes can be limited to values below 360° with M94.

Circular Movement, Cartesian Interpolation planes



Main planes



Circular arcs can be directly programmed in the main planes XY, YZ, ZX.

The circular interpolation plane is selected by defining the spindle axis with "T". This also assigns the tool compensations.

The axis printed bold below (e.g. **X**) is identical in its positive direction with the angle 0° (leading axis). The axis in normal print points in the 90° direction.

Interpolation planes

Spindle axis parallel to	Circular interpolation plane	
Z	XY	
Y	ZX	
X	YZ	

Oblique circles in space

Circular arcs which are not parallel to a main plane can be programmed via Q parameters and executed as a sequence of multiple short straight lines (G01 blocks).

Circular Movement, Cartesian

Selection guide: Arbitrary transitions

G02/G03 and G05



Circular movement

The control moves two axes simultaneously, so the tool describes a circular arc relative to the workpiece.

Arbitrary transitions

The functions G02 and G03 define – together with the preceding block – arbitrary transitions at the beginning and end of the arc.

Difference between G02/G03 and G05

If a program section contains a contour which has to be programmed as alternating linear and circular movements, the G05 function can be used while still retaining the direction of rotation programmed via G02 or G03. G05 corresponds in function and input to the functions G02/G03. The only difference is that with G05 you do not need to enter the direction of rotation. That is, G05 generates both clockwise (CW) and counterclockwise (CCW) circular movements. The prerequisite for employing G05 is that the direction of rotation has previously been programmed via G02/G03.

Prerequisite

The starting point ① of the circular movement must be approached in the immediately preceding block.

Circle endpoint

The circle endpoint ② is programmed in a G02 or G03 block.

Direction of rotation G02/G03

In mathematical terms, the **negative** direction of rotation "G02" is **clockwise** (CW).

The **positive** direction of rotation "G03" is **counterclockwise**.

Radius

For G02/G03, the radius results from the distance of the position immediately before the block which was programmed with G02/G03 (beginning of circle) to the circle center I, J, K.

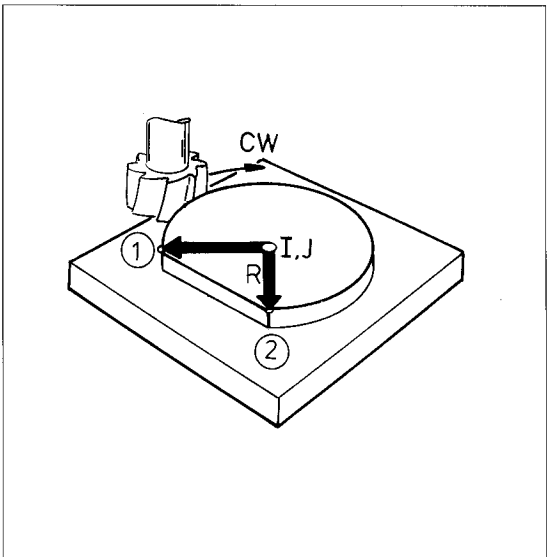
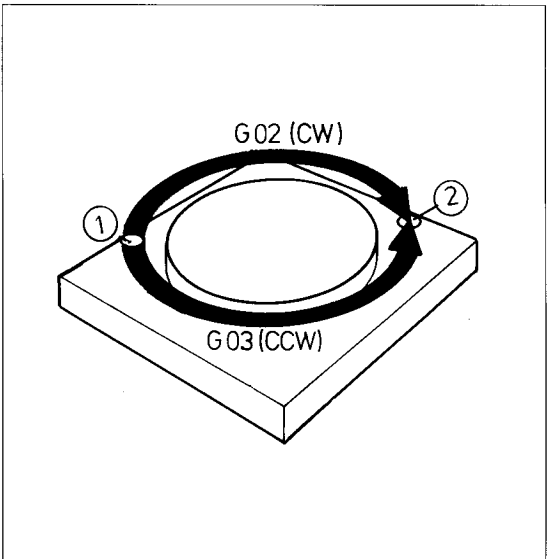
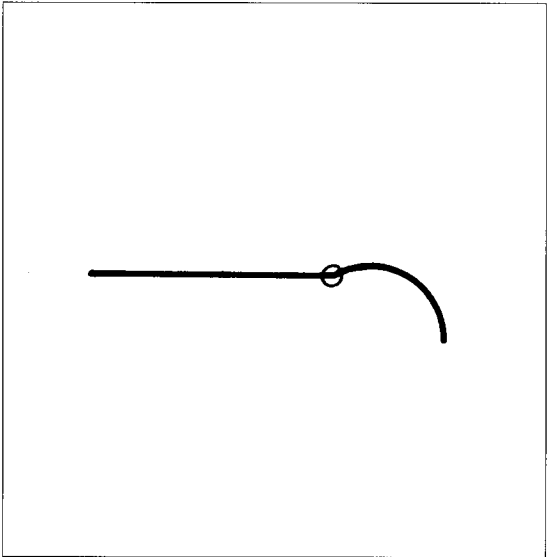
Full circles

A full circle can be programmed in one block only with G02/G03.

You can enter the radius directly with "R" (without I, J, K).

Selection:

Given	Required path function
Arc starting point ①	e.g. G01 traverse to the starting point
Circle center	I, J, K
Arc end point ②	G02/G03
Arc starting point ①	e.g. G01 traverse to the starting point
Radius + arc end point ②	G02/G03 mit Radius R



Circular Movement, Cartesian

Selection guide: Tangential transitions

G25/G06

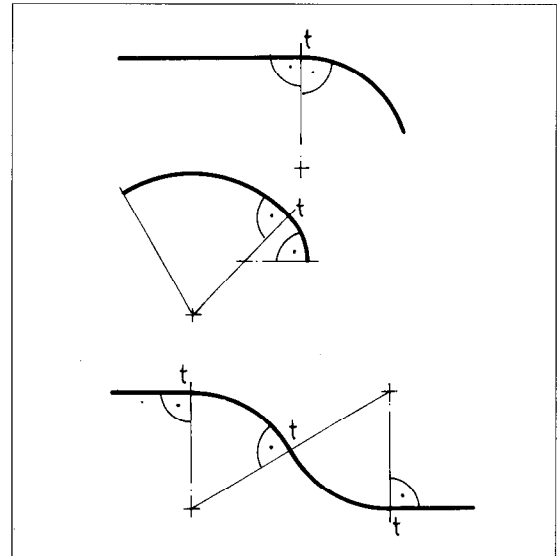


Tangential transitions

The G25 and G06 functions automatically produce a tangential (soft) entry into the arc. Departure from the arc is also tangential with G25, and arbitrary with G06. The direction of movement when entering the circle thus also determines the shape of the arc.

Direction of rotation

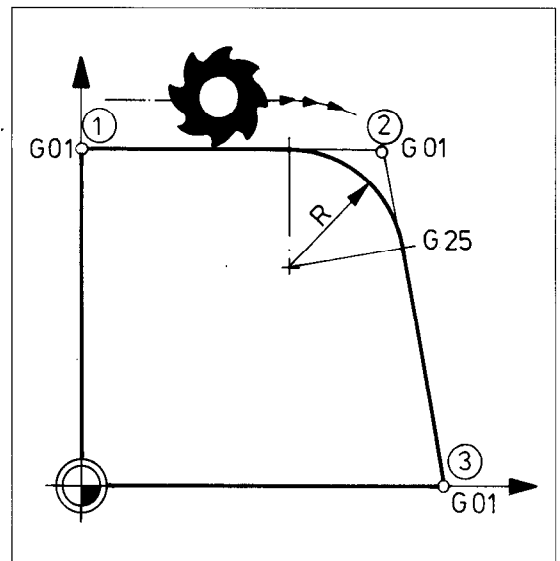
The direction of rotation need therefore not be given.



Corner rounding: G25

Corner rounding with G25 is inserted between two contour elements which can be straight lines or arcs.

The data to be programmed are: the **corner point** ② (which is not traversed), and directly following it a separate rounding block G25 with the rounding radius R. Entry into and exit from the rounding radius is tangential and is automatically computed by the control.

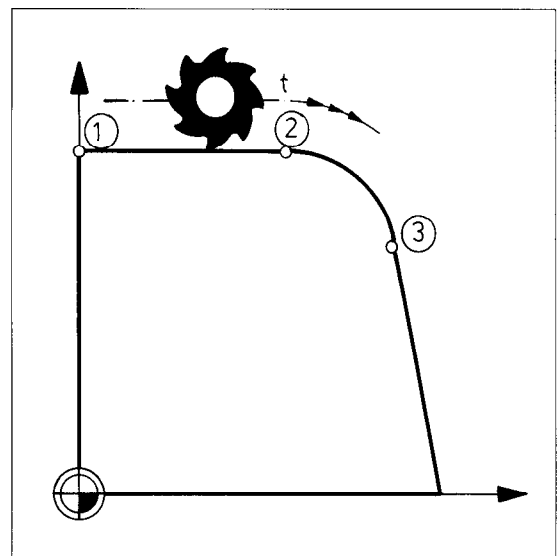


Tangential contour connection G06

With G06 **only the arc end point** ③ is programmed.

Selection:

Given	Required path function
Point ①	Traverse e.g. with G01
"Corner" ②	Traverse e.g. with G01
Rounding radius	G25
Point ③	Traverse e.g. with G01
Tangent generating point ①	Traverse e.g. with G01
Tangential arc ②	Traverse e.g. with G01
Arc end point ③	G06



Circular Movement, Cartesian

Arc with circle center: I, J, K + G02/G03



I J K

I, J and K have two functions:

1. Specifying the **circle center** for circular arcs with G02/G03.
2. Defining the **pole** as datum for position data in polar coordinates.

Circle center I, J, K

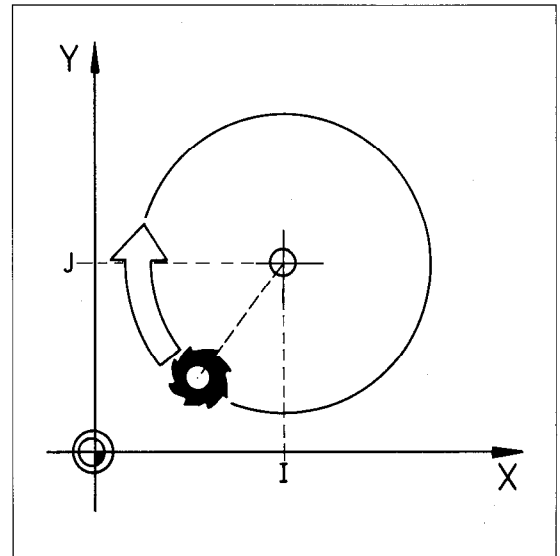
The circle center I, J, K must be determined before circular interpolation with G02/G03 and may be programmed in one block with the circular movement. This circle center remains in effect until replaced by a new I, J, K command.

There are three methods for programming:

- The circle center I, J, K is directly defined by Cartesian coordinates.
- The coordinates last programmed in a I, J, K block define the circle center.
- The current position is taken as circle center with G29 (without numerical input).

This is also possible for positions programmed in polar coordinates.

The circle center is defined by two coordinates in the working plane:



Working plane
(Circular interpolation
plane)

Circle center
coordinates

X Y
Z X
Y Z

I, J
K, I
J, K

I, J, K absolute: the circle center is based on the work datum.

I, J, K incremental: the circle center is based on the tool position last programmed.

Programming in the circle center produces no movement!

Approaching the starting point

Approach the starting point for the circular arc before the G02/G03 block.

Radius

The distance from the starting point to the circle center determines the radius.

Circular arc G02/G03

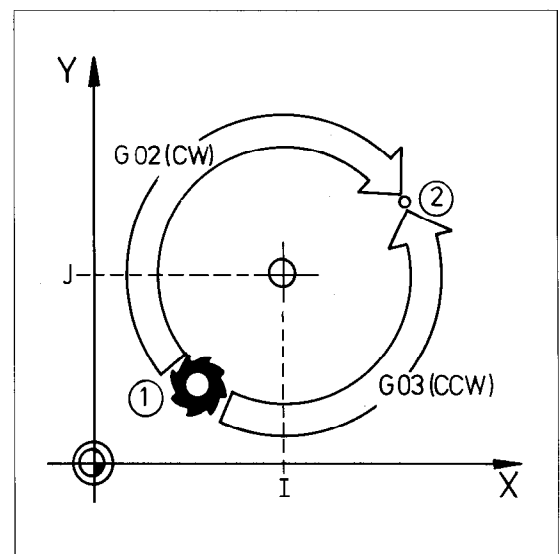
The tool is to travel from position ① to target point ② in a circular path. Only program ② in the G02/G03 block. Position ② can be entered in Cartesian or polar coordinates.

Direction of rotation

The direction of rotation must be defined for circular movement:
rotation in negative direction **G02** (clockwise).
rotation in positive direction **G03** (counterclockwise)



Any tool radius compensation must begin before a circular arc.



Circular Movement, Cartesian

Arc with circle center: I, J, K + G02/G03



Input tolerance

The starting and endpoint must lie on the same circular path, i.e. they must be at the same distance from the circle center CC. The tolerance of position inputs for the starting position, end position and circle center is $\pm 8 \mu\text{m}$.

Input circle center

I +50 **J** +50

Circle center

Input G02/G03

G 02 **X** +15 **Y** +50 **END**

Specify the rotating direction with G02: (direction of rotation clockwise) and arc end point.

Program blocks

I+50 J+50 *
G02 X+15 Y+50 *

G41/G42, F and M are entered as for straight lines. They are only necessary when different from previous input.

Example full circle

Full circle in the XY plane
(outer circle) around center
X+50, Y+50 with 35 mm radius.

Program

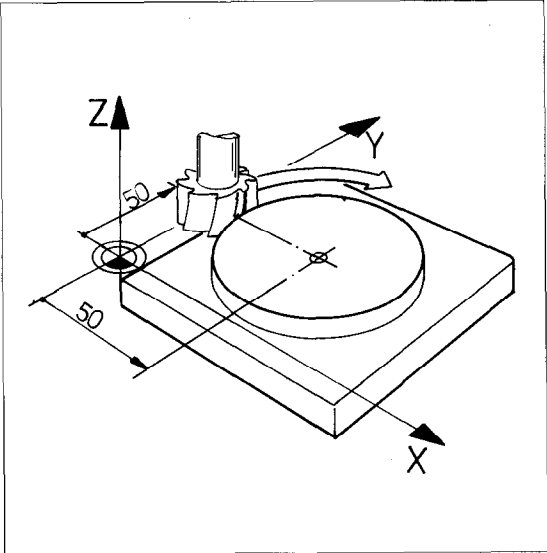
G99 T1 L+0 R5 *
T1 G17 S200 *

G01 G41 X+15 Y+50 F300 M03 *

I+50 J+50 *

G02 X+15 Y+50 *

Full circles can be programmed with G02/G03 in one block.
The circle starting point and the circle endpoint are identical.



Example arc

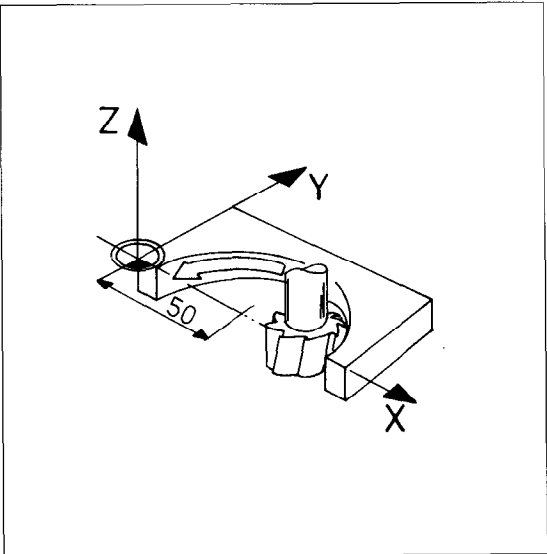
Semicircle in the XY plane
(inside circle) around center
X+50 Y+50 with 35 mm radius.

Program

G01 G41 X+85 Y+50 F300 M3 *

I+50 J+0 *

G03 X+15 Y+50 *



Circular Movement, Cartesian

Corner rounding with radius: G02/G03



Circular arc G02/G03

If the contour radius is given in the drawing, but no circle center, the circle can be defined via G02/G03 key with the

- endpoint of the circular arc
- radius and
- direction of rotation.

G41/G42, F and M are entered as for straight lines and are only required when changing earlier specifications.

Starting point

The starting point of the arc must be approached in the preceding block.

Endpoint

In the G02/G03 block the endpoint can only be programmed with Cartesian coordinates.

The distance between starting and end point of the arc must not exceed $2 \times R$! With G02/G03, full circles can be programmed in 2 blocks.

Central angle

There are two geometric solutions for connecting two points with a defined radius (see figure), depending on the size of the central angle β : The smaller arc 1 has a central angle $\beta_1 < 180^\circ$, the larger arc 2 has a central angle $\beta_2 > 180^\circ$.

Contour radius

Enter a **positive** radius to program the smaller arc ($\beta < 180^\circ$).

(The + sign is automatically generated.)

To program the larger arc ($\beta > 180^\circ$), enter the radius as a **negative** value.

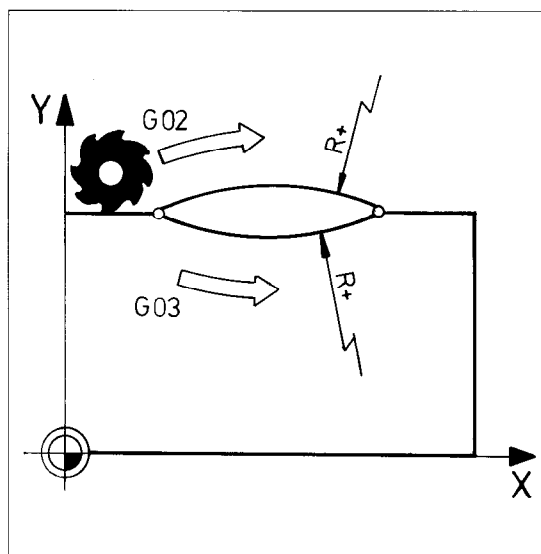
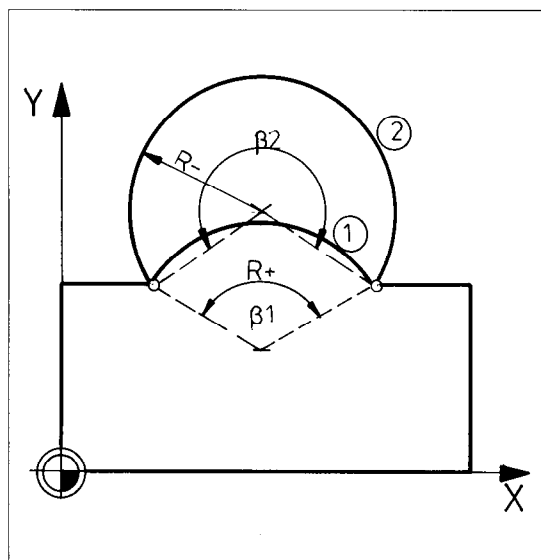
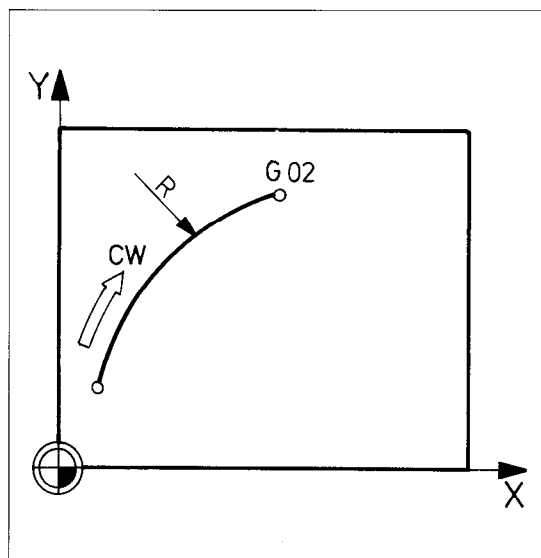
The maximum definable radius = 30 m.

Arcs up to 99 m can be produced with parametric programming.

Rotating direction

Depending on the allocation of radius compensation G41/G42, the rotating direction determines whether the circle curves inward (= concave) or outward (= convex).

In the adjacent figure, G02 produces a convex contour element, G03 a concave contour element.



Circular Movement, Cartesian

Corner rounding with radius: G02/G03



Input G02



X 80 Y 40

R 100

Circle, Cartesian, clockwise

Endpoint of arc

Radius, positive sign

Program block

G02 X+80 Y+40 R+100 *

Examples:

G99 T1 L+0 R+5 *
T1 G17 S200 *

Arc A

G01 G41 X+20 Y+60 F300 M03 *

G02 X+80 Y+60 R+50 *

Arc B

G01 G41 X+20 Y+60 F300 M03 *

G02 X+80 Y+60 R-50 *

Arc C

G01 G41 X+20 Y+60 F300 M03 *

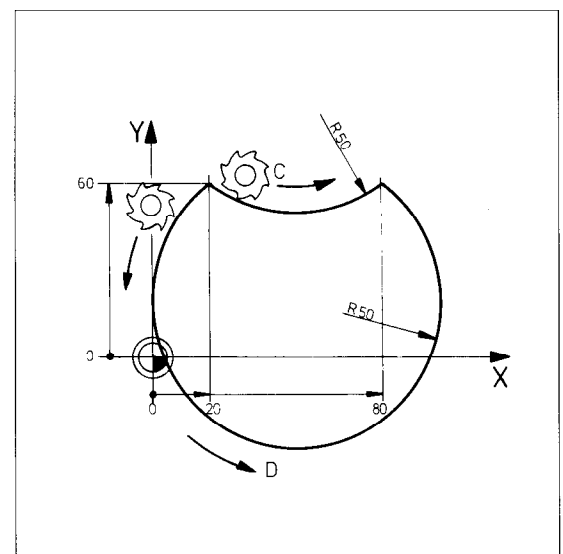
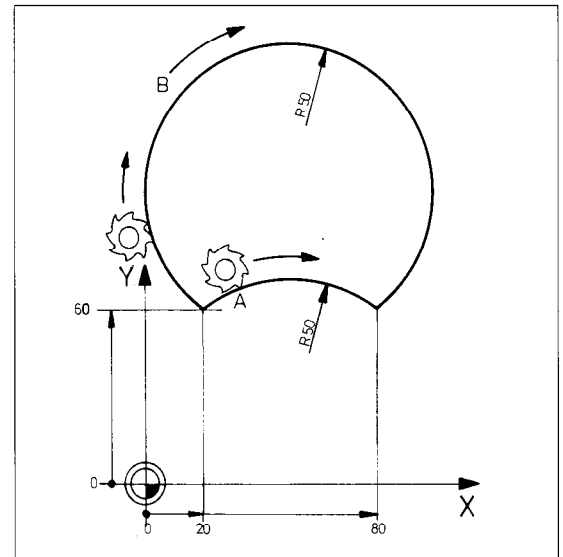
G03 X+80 Y+60 R+50 *

Arc D

G01 G41 X+20 Y+60 F300 M03 *

G03 X+80 Y+60 R-50 *

The position X+20 Y+60 is the start of arc in the examples; the position X+80 Y+60 is the end of arc.



Circular Movement, Cartesian

Corner rounding with radius R: G25



Circular arc G25

Contour corners can be rounded with arcs. The circle connects tangentially with the preceding and succeeding contour.

A rounding arc can be inserted at any corner formed by the intersection of the following contour elements:

- straight line – straight line,
- straight line – circle, or circle – straight line,
- circle – circle.

Prerequisites

Rounding is completely defined by the G25 block and the points ① ② ③. A positioning block containing both coordinates of the machining plane should be programmed before and after the G25 block. The G40/G41/G42 compensation must be identical before and after the G25 block.

A contour therefore cannot be started in a corner which is to be rounded.

Note

The rounding arc can only be executed in the machining plane. The machining plane must be the same in the positioning block before and after the rounding block.

The rounding radius cannot be too large or too small for inside corners – it must “fit between the contour elements” and be machinable with the current tool.

The feed rate for corner rounding is effective blockwise. The previously programmed feed rate is reactivated after the G25 block.

Programming

The rounding arc is programmed as a separate block following the corner to be rounded. Enter the rounding radius and a reduced feed rate F, if needed. The “corner point” itself is not traversed!

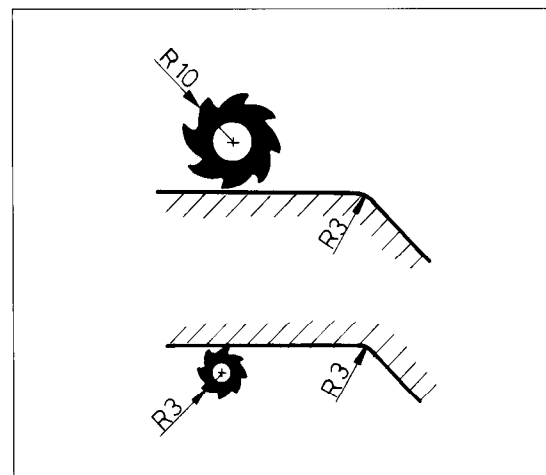
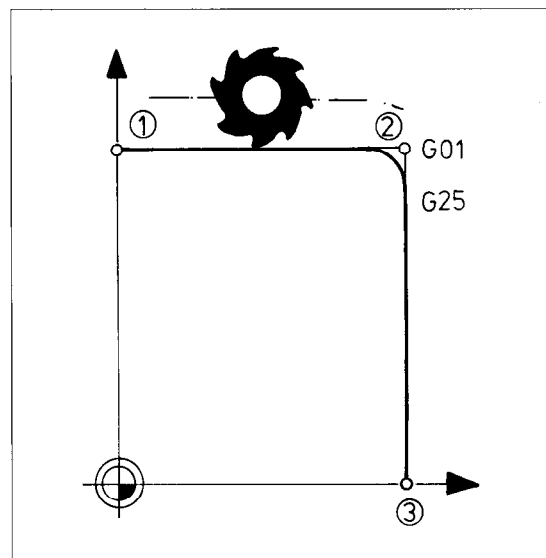
Error messages

PLANE WRONGLY DEFINED

The machining planes are not identical before and after the RND block.

ROUNDING RADIUS TOO LARGE

The rounding is geometrically impossible.



The tool radius can be larger than the rounding radius on outside corners.

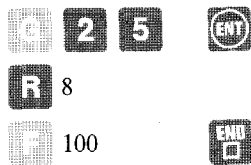
The tool radius must be smaller than or equal to the rounding radius on inside corners.

Circular Movement/Cartesian

Corner rounding with radius R: G25



Input G25



Corner rounding

Rounding radius

A separate feed rate can be entered and is only effective for this rounding.

Program block

G25 R8 F100 *

Examples:

G99 T1 L+0 R+5 *
T1 G17 S200 *

Sequence A

G01 G41 X+10 Y+60
F300 M03 *

X+50 Y+60 *

G25 R7 *

X+90 Y+50 *

Position ①
"Corner point" ②
Rounding
Position ③

Sequence B

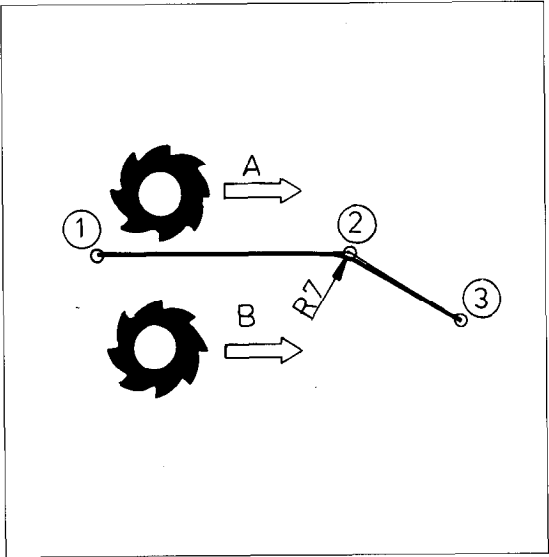
G01 G42 X+10 Y+60
F300 M03 *

X+50 Y+60 *

G25 R7 *

X+90 Y+50 *

Position ①
"Corner point" ②
Rounding
Position ③



Circular Movement, Cartesian

Tangential arc with end point X, Y: G06



Circular arc G06

A circular arc can be programmed more easily if it connects tangentially to the preceding contour. The circular arc is defined by **merely entering the arc endpoint** with G06.

Geometry

An arc with tangential connection to the contour is exactly defined by its endpoint.

This arc has a specific radius, a specific direction of rotation and a specific center. This data need not therefore be programmed.

Prerequisites

The contour element which connects tangentially to the circle is programmed immediately before the tangential arc. Both coordinates of the same machining plane must be programmed in the block for the tangential arc and in the preceding block.

Tangent

The tangent is specified by **both** positions ① and ② directly preceding the G06 block. Therefore, the first G06 block can appear no earlier than the third block in a program.

Path of the circular arc G06

The tool is to travel a circle connecting tangentially to ① and ② to target point ③. Only ③ is programmed in the G06 block.

Coordinates

The endpoint of the circular path can be programmed in either Cartesian or polar coordinates.

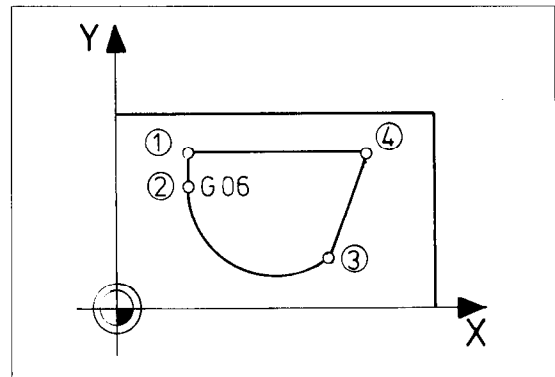
Error messages

WRONG CIRCLE DATA

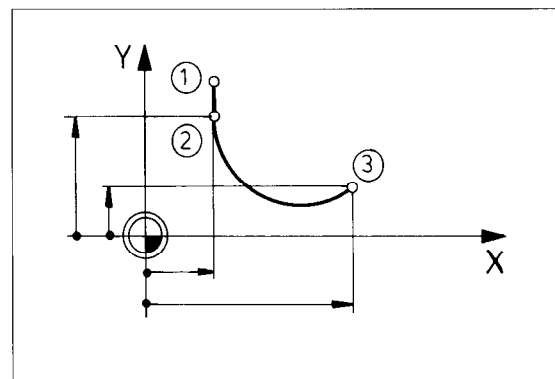
The required minimum 2 positions before the G06 block were not programmed.

ANGLE REFERENCE MISSING

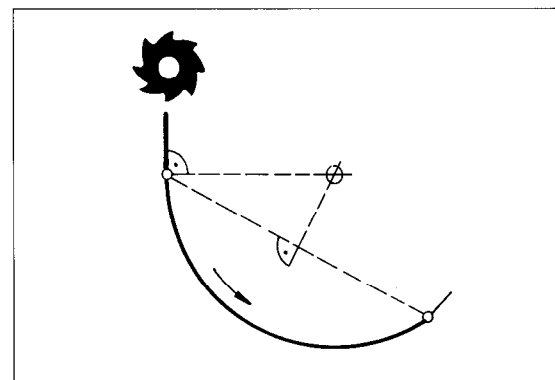
Both coordinates of the machining plane are not given in the G06 block and the preceding block.



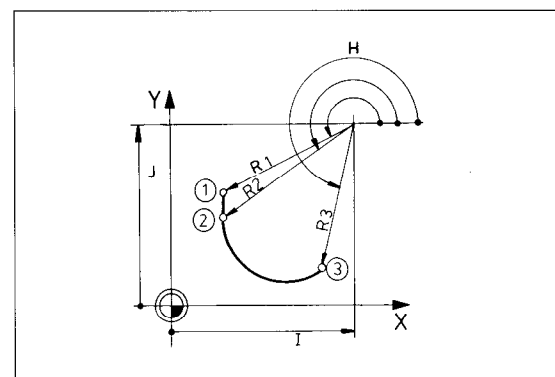
Machining sequence



Geometry



Cartesian coordinates



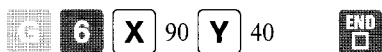
Polar coordinates

Circular Movement, Cartesian

Tangential arc with end point X, Y: G06



Input G06



Arc endpoint

Program block

G06 X+90 Y+40 *

Enter R, F and M as for straight lines.
Input is only necessary to change earlier definitions.

**Examples:
different
endpoints**

G30 G17 X+0 Y+0 Z-40 *
G31 G90 X+130 Y+100 Z+0 *
T1 G17 S200 *

Arc A

G01 G41 X+10 Y+80
F300 M03 *
X+50 *
G06 X+130 Y+30 *

1st tangent point
Start of arc
End of arc.

**Arc B
semicircle**

G01 G41 X+10 Y+80
F300 M03 *
X+50 *
G06 X+50 Y+0 *

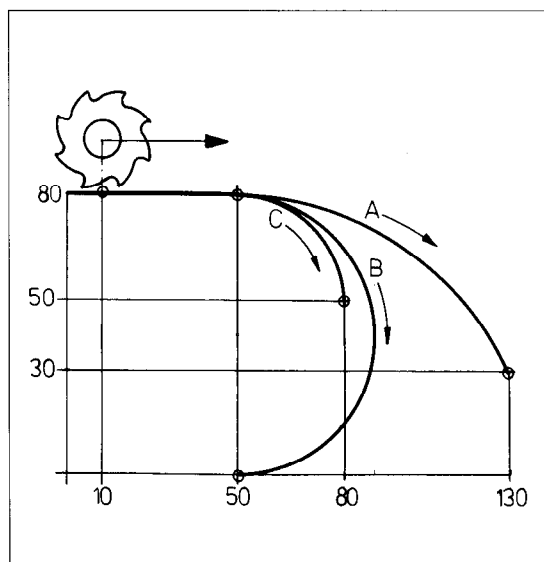
1st tangent point
Start of arc
End of arc.

A semicircle with
R = 40 is formed.

**Arc C
quarter circle**

G01 G41 X+10 Y+80
F300 M03 *
X+50 *
G06 X+80 Y+50 *

1st tangent point
Start of arc
End of arc.
A quarter circle with
R = 30 is formed.



**Different
tangents**

Arc A

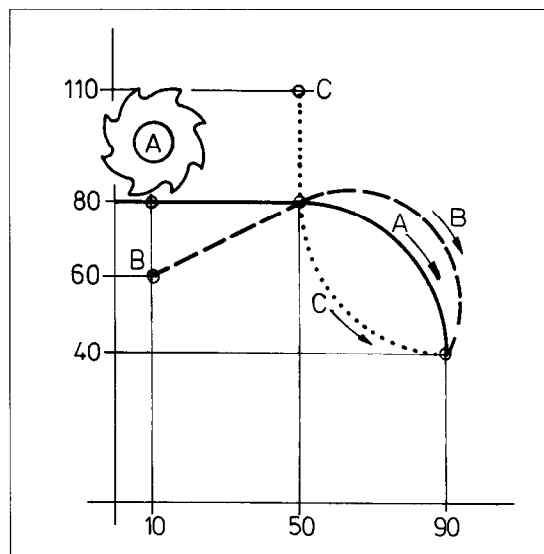
G01 G41 X+10 Y+80 F300 M03 *
X+50 *
G06 X+90 Y+40 *

Arc B

G01 G41 X+10 Y+60 F300 M03 *
X+50 Y+80 *
G06 X+90 Y+40 *

Arc C

G01 G41 X+50 Y+110 F300 M03 *
Y+80 *
G06 X+90 Y+40 *



Polar Coordinates

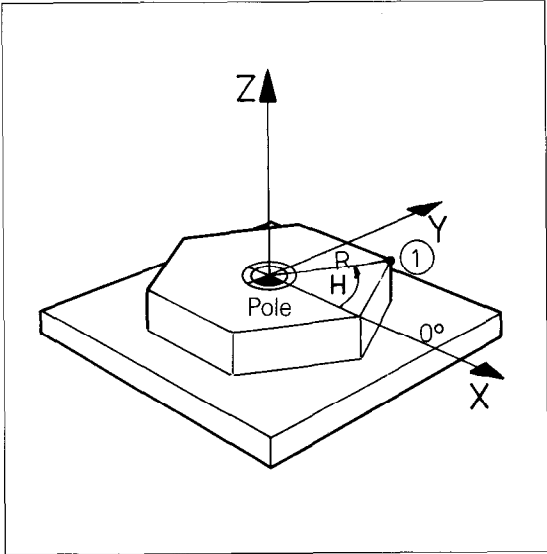
Fundamentals



The control also allows you to enter nominal positions in polar coordinates.

In polar coordinates, the points in a plane are specified by the **polar radius R** (distance to the pole), and the **polar angle H** (angular direction).

The pole position is entered with the I, J, K keys in Cartesian coordinates based on the workpiece datum.

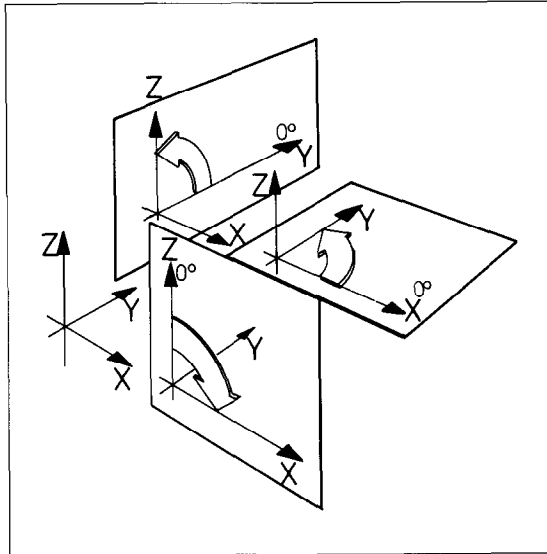


Angle reference axis

The angle reference axis (0° axis) is the +X axis in the XY plane, +Y axis in the YZ plane, +Z axis in the ZX plane.

The machining plane (e.g. XY plane) is determined by a tool call.

The sign of the angle H can be seen in the adjacent figure.



Absolute polar coordinates

Absolute dimensions are based on the current pole.
Example: G11 G90 R+50 H+40 *

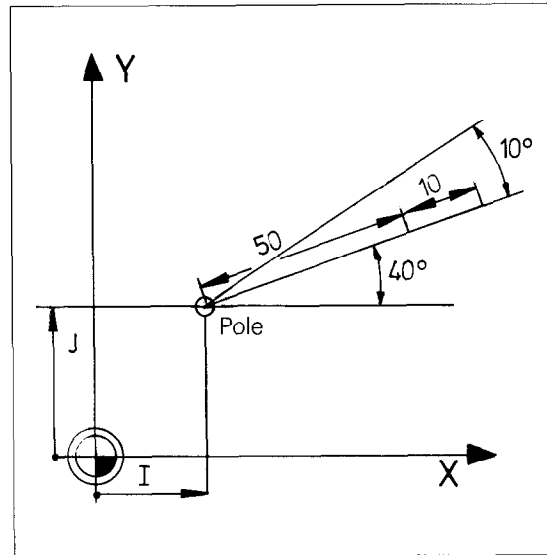
Incremental polar coordinates

A polar coordinate radius entered incrementally changes the last radius.
Example: G11 G91 R+10 *

An incremental polar coordinate angle IPA refers to the last direction angle.
Example: G11 G91 H+15 *

Mixing

Absolute and incremental coordinates may be mixed within one block.
Example: G11 G90 R+50 G91 H+15 *



Polar Coordinates

Pole: I, J, K



Pole



Before entering polar coordinates, the pole has to be defined with I, J, K. The pole can be defined at any point in the program **before** the first application of polar coordinates.

The pole is programmed in Cartesian coordinates, either as absolute or incremental dimensions.

Pole in absolute dimensions: The pole is referenced to the workpiece datum.

Pole in incremental dimensions: The pole is referenced to the last-programmed nominal position of the tool.

The coordinates of the pole are determined by the working plane:

Working plane	Polar coordinates
X Y	I, J
Y Z	J, K
Z X	K, I

Example

I+60 J+30 *

Transferring the pole G29

The last **programmed position** is **transferred** as the pole with G29.

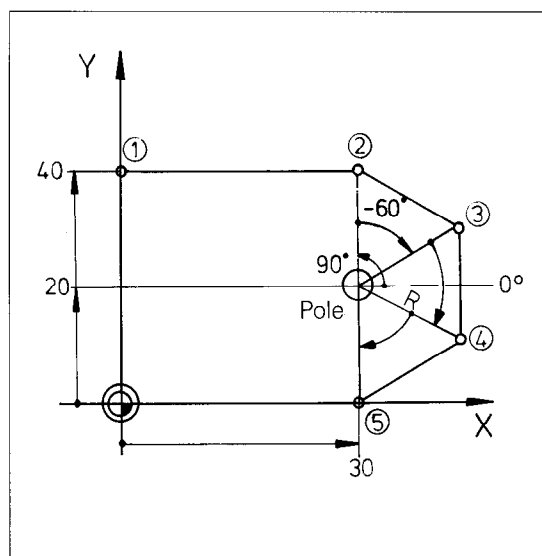
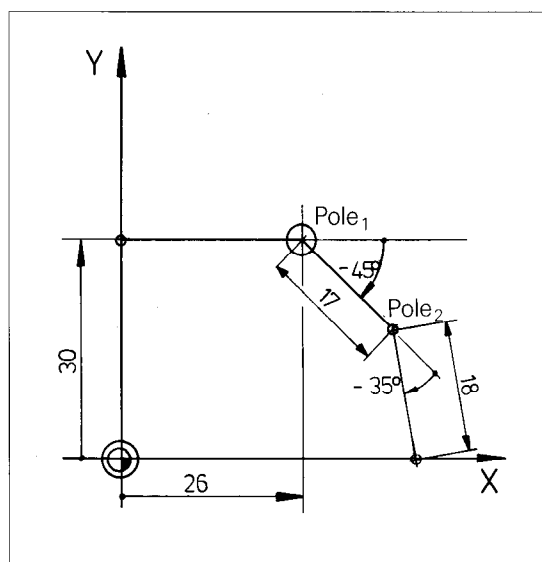
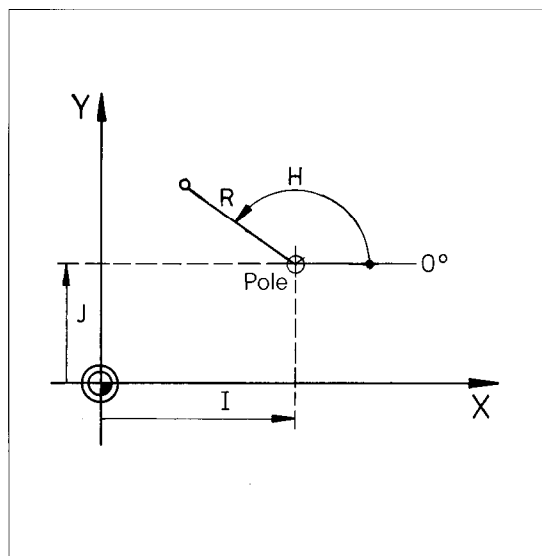
Directly transferring the pole in this manner is especially well suited for polygon shapes (see illustration at right).

Example

G01 X+26 Y+30
G29 POLE 1
G11 R+17 H-45
G29 POLE 2
G11 R+18 G91 H-35

Modal effect

A pole definition remains valid in a program until it is overwritten with another definition. The same pole therefore need not be programmed repeatedly.



Polar Coordinates

Straight lines: G10/G11



G10/G11

For dimensions which are referenced to a rotational axis in some way, programming is usually easier in polar coordinates than in Cartesian coordinates because calculations are avoided.

Range for polar angle H

Input range for linear interpolation: absolute or incremental -360° to $+360^\circ$.

H positive: counterclockwise angle.

H negative: clockwise angle.

Example

Milling an inside contour:

Program

```
G30 G17 X+0 Y+0 Z-40 *
G31 G90 X+100 Z+0 *
G99 T2 L+0 R+2 *
T2 G17 S200 *
```

```
I+50 J+60 *
```

Set POLE^{*)}

```
G01 G40 G90 X+15
```

Approach starting point externally (Cartesian coordinates)

```
Z-5 F100 *
```

Plunge

```
G11 G42 R+40
H+180 F200 *
```

Approach 1st contour point with compensation (polar coordinates)
2nd contour point

```
G91 H-60 *
```

```
H-60 *
```

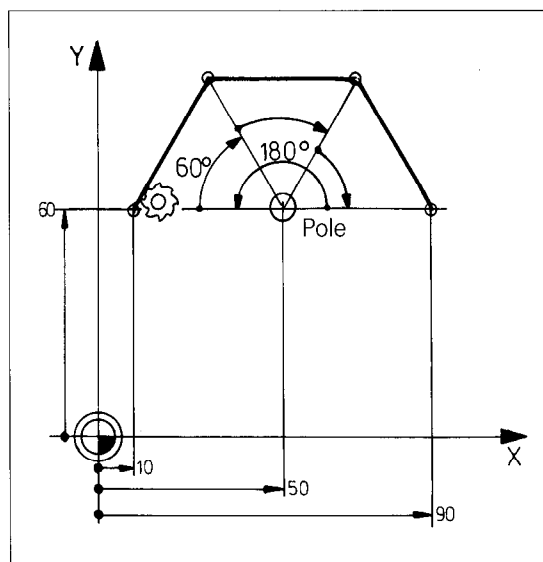
```
H-60 *
```

```
G40 G90 X+85 Y+50 *
```

Last contour point
Depart from contour, cancel compensation

```
G00 Z+50 M02 *
```

Retract, return jump to beginning of program



^{*)} The pole can also be programmed in the block with G11.

Polar Coordinates

Circular arcs: G10/G11



Circular arc G12/G13



If the target point on the arc is programmed in polar coordinates, you only have to enter the polar angle H to define the endpoint. The radius is defined by the distance from the starting point of the arc to the programmed circle center I, J, K.

When programming a circle in polar coordinates, the angle H can be entered positively or negatively. The angle H determines the endpoint of the arc.

If the angle H is entered incrementally, the sign of the angle and the sign of the rotating direction should be the same. In the figure to the right, this means that H is negative and the direction of rotation is also negative (G12).

Range for polar angle

Input range for circle interpolation:
absolute or incremental -5400° to $+5400^\circ$.

Example

An arc with radius 35 and circle center X+50 Y+60 is to be milled.
Rotating direction is clockwise.

Program

```
G99 T1 L+0 R5 *  
T1 G17 S200 *
```

```
I+50 J+60 *
```

Coordinates of
circle center

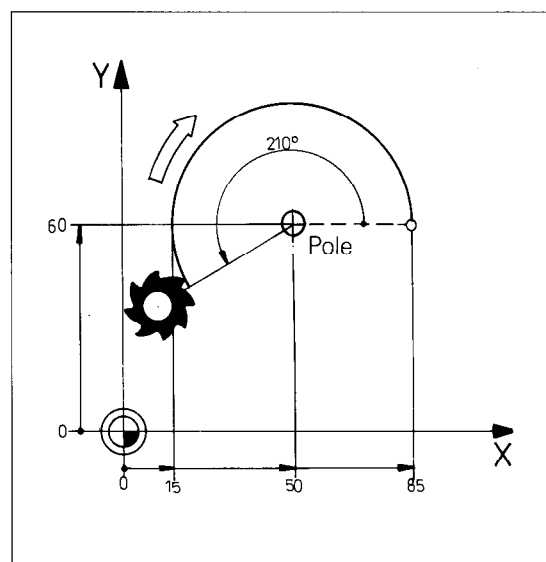
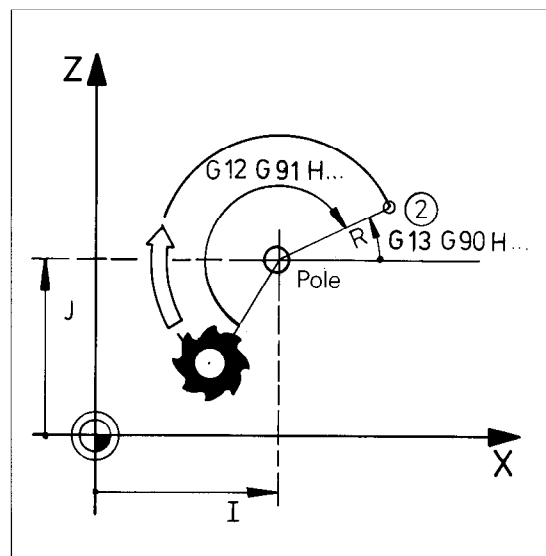
```
Z-5 F100 *
```

Plunge

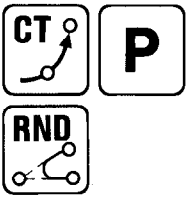
```
G11 G41 R+35 H+210 F200 M3 *Approach circle  
(circle radius is 35 mm)
```

```
G12 H+0 F300 *
```

Circular
movement
clockwise



In the example, a contour radius of 35 mm is obtained from the distance between the POLE and the approach point on the circle.



Polar Coordinates

Tangential arcs: G16

Corner rounding: G25



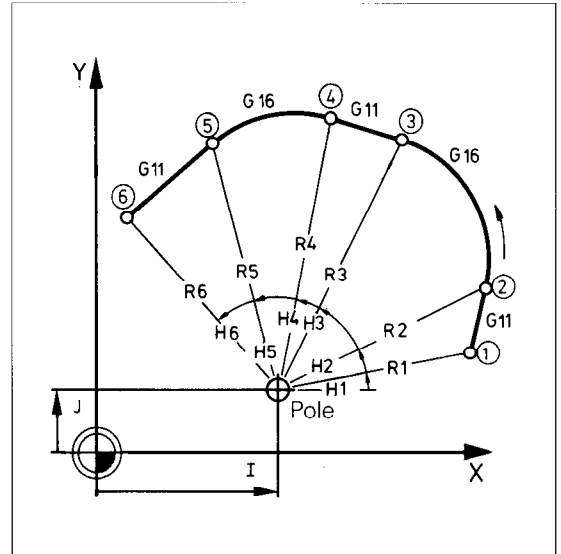
Tangential arc G16

The endpoints of tangential arcs may be entered in polar coordinates to simplify the programming of, for example, cams.

The start of the arc is automatically tangential when programming with G16.

If the transition points are not calculated exactly, the arc elements could become "jagged".

Specify the pole CC before programming in polar coordinates.



Example

A straight line through ① and ② is to tangentially meet the arc to ③. The radius and direction angle of ③ with respect to I, J, K, are known.

Program

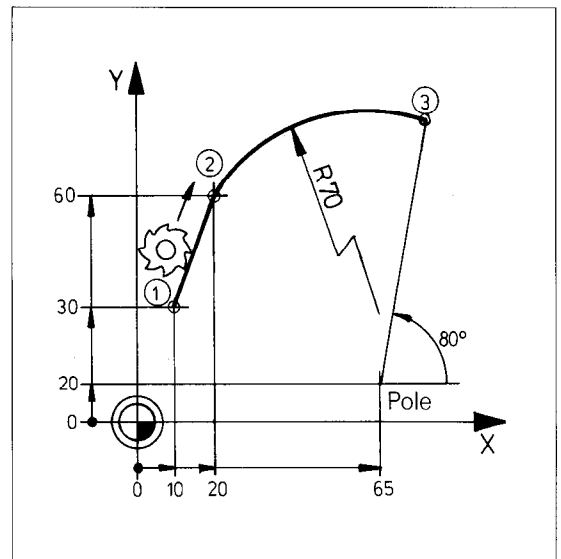
```
G99 T1 L+0 R4 *
T1 G17 S200 *
```

```
I+65 J+20 *
```

```
G01 G41 X+10 Y+30 F500 M03 *
```

```
X+20 Y+60 *
```

```
G16 R+70 H+80 *
```



G25

Polar "corners" can also be rounded with the "corner rounding" function (see Circular Movement, Cartesian, Corner rounding).

Polar Coordinates

Helical interpolation: I, J, K + G12/G13



Helix

If 2 axes are moved simultaneously to describe a circle in a main plane (XY, YZ, ZX), and a uniform linear motion of the tool axis is superimposed, then the tool moves along a helix (helical interpolation).

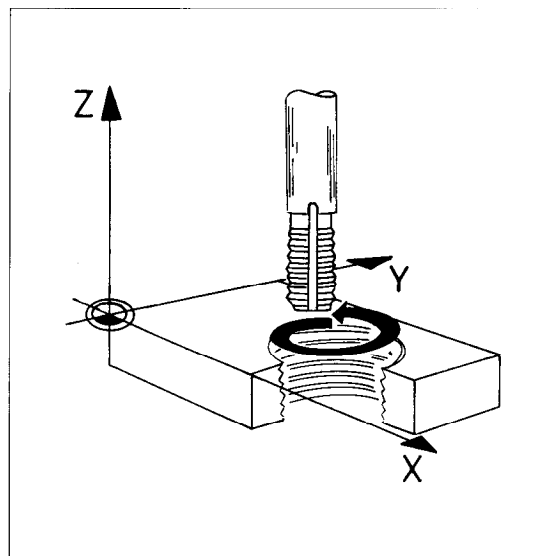
Applications

Helical interpolation can be used to advantage with form cutters for producing internal and external threads with large diameters, or for lubricating grooves. This can save you substantial tool costs.

Input data

The helix is programmed in polar coordinates.

First specify the POLE or circle center (e.g. I, J).



Angle range

Enter the total angle of tool rotation for the polar angle H in degrees:

$H = \text{number of rotations} \times 360^\circ$

Maximum angle of rotation: $\pm 5400^\circ$ (15 complete rotations).

Height

The total height L (= Z) is entered for the tool axis.

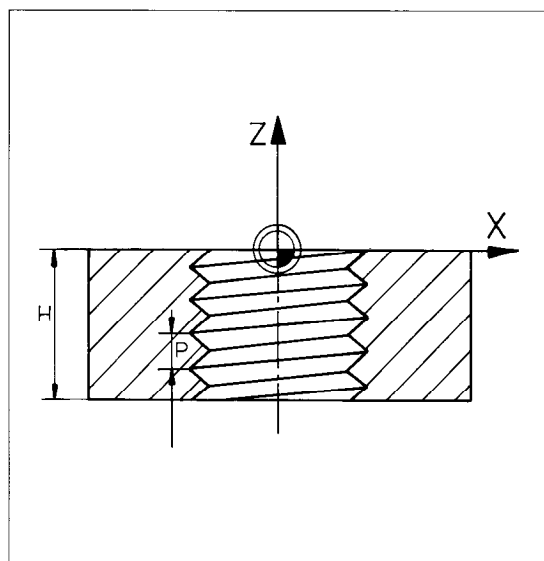
Calculate the value from the thread pitch and the required number of tool rotations.

$Z = P \cdot n$, $Z = \text{total height/depth to be entered}$

$P = \text{pitch}$

$n = \text{number of threads}$

The total height/depth can be entered in absolute or incremental dimensions.



Thread

A complete thread can be programmed quite easily with Z and H; the number of threads is then specified with a program section repeat REP.

Radius compensation

The radius compensation depends upon the

- rotating direction (right/left),
- type of thread (internal/external),
- milling direction (positive/negative axis direction)
(see table to the right).

Internal thread	Working direction	Rotating direction	Radius compensation
right-hand	Z+	G13	G41
left-hand	Z+	G12	G42
right-hand	Z-	G12	G42
left-hand	Z-	G13	G41

External thread	Working direction	Rotating direction	Radius compensation
right-hand	Z+	G13	G42
left-hand	Z+	G12	G41
right-hand	Z-	G12	G41
left-hand	Z-	G13	G42

Polar Coordinates

Helical interpolation: I, J, K + G12/G13



Input example

C **1** **3**

Circle, polar,
counterclockwise

C **9** **1**

H **3** **6** **0** **Z** **2** **END** Endpoint

G13 G91 H+360 Z+2 *

Task

A right-hand internal thread M64 x 1.5 is to be produced in one cut with a multi-cutter tool.

Thread

Thread data:

pitch

$P = 1.5 \text{ mm}$

start

$\alpha_a = 0^\circ$

end

$\alpha_e = 0^\circ = 360^\circ$

Number of threads

$n_0 = 5$

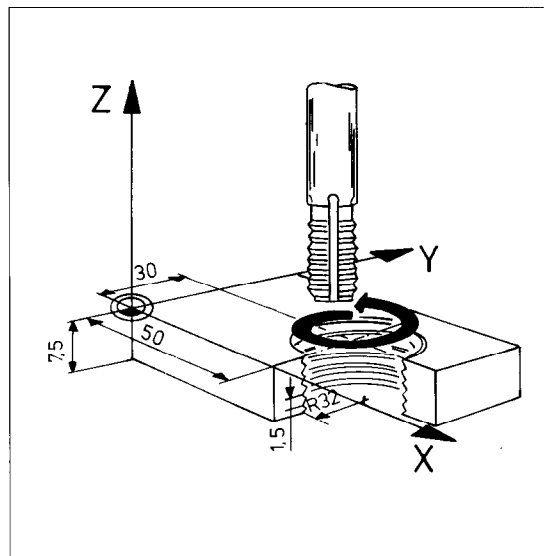
Overrun of threads:

at start

$n_1 = 1/2$

at end

$n_2 = 1/2$



Calculations

Total height:

$Z = P \cdot n = 1.5 \text{ mm} \cdot [5 + (2 \cdot 1/2)] = 9 \text{ mm}$

Incremental polar angle:

$H = 360^\circ \cdot n = 360^\circ \cdot [5 + (2 \cdot 1/2)] = 2160^\circ$

Due to overrun of 1/2 thread, the start of thread is advanced by 180°:

starting angle $\alpha_s = \alpha_a + (-180^\circ) = 0^\circ + (-180^\circ) = -180^\circ$

The overrun of 1/2 thread at the start of thread gives the following initial value for Z:

$Z = -P \cdot n = -1.5 \text{ mm} \cdot [5 + 1/2] = -8.25 \text{ mm}$

Program

```
%20 G71 *
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T1 L+0 R+5 *
N40 T1 G17 S500 *
N50 G00 G90 Z+200 M06 *
N60 G00 G40 X+50 Y+30 *
N65 G29 *
N70 Z-8.25 M03 *
N75 G11 G41 R+32 H-180 F100 *
N80 G13 G91 H+2160 Z+9 F200 *
N90 G01 G40 X+50 Y+30 *
N95 G00 Z+200 M02 *
N9999 %20 G71 *
```

Workpiece blank definition

Tool definition

Tool call

Move to the tool change position

Move to hole center

Define position as pole

Downfeed in center

Move to wall

Helical movement

Retract in XY

Retract in Z

Note

Helical interpolation cannot be graphically displayed.

Contour Approach and Departure

Starting and end position



Selecting the 1st contour point

Before beginning contour programming, specify the first contour point at which machining **with radius compensation** is to begin.

Starting point

In the vicinity of the first contour point, define an **uncompensated starting point** that can be approached in rapid traverse, and be sure to consider the tool in use. The starting point must fulfill the following criteria:

- approachable without collision
- near the first contour point
- outside the material
- the contour will not be damaged when approaching the first contour point.

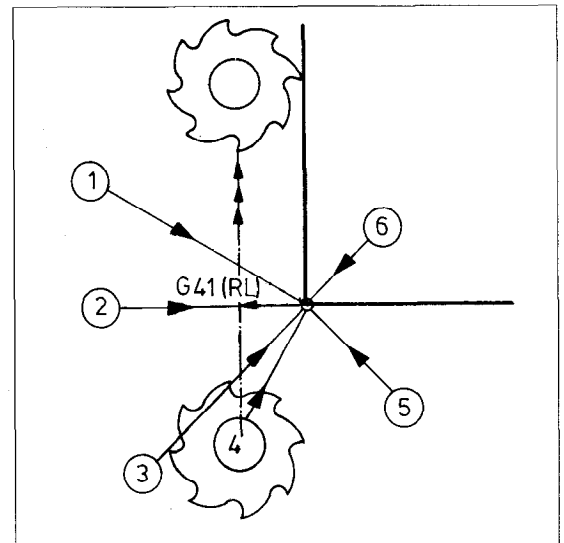
Direct approach

When working on a circle (G26/G27) without the TNC approach/departure function, also check that the tool does not blemish the contour due to a direction change.

Starting points

- | | |
|---|--|
| <p>① Not recommended</p> <p>② Not recommended</p> <p>③ Suitable</p> <p>④ Optimal</p> <p>⑤ Not recommended</p> <p>⑥ Not permitted!</p> | <p>Surface blemish due to change of Y-axis direction</p> <p>Also for end point</p> <p>Lies on the extension of the compensated path</p> <p>Contour damage</p> |
|---|--|

Radius compensation must remain switched off for the starting position (G40).



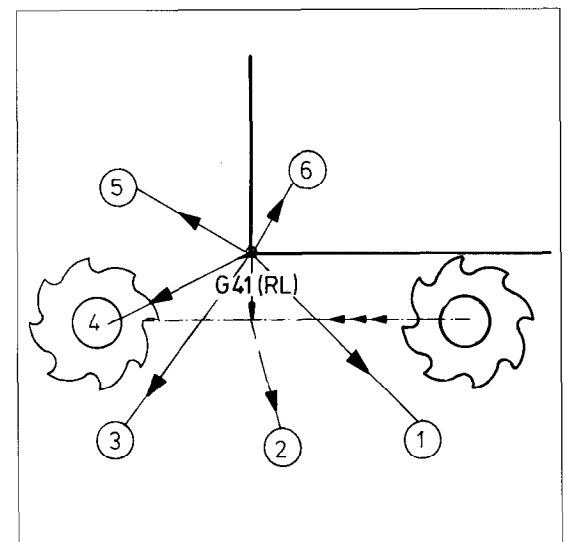
End points

The same prerequisites apply for selecting the **uncompensated end point** as for the starting point.

The ideal end point ④ lies on the extension of the last contour element G41.

- | | |
|---|---|
| <p>①, ② Not recommended</p> <p>③ Suitable</p> <p>④ Optimal</p> <p>⑤ Not recommended</p> <p>⑥ Not permitted!</p> | <p>Surface blemish due to change of the X-axis direction</p> <p>Also for the starting point</p> <p>Lies on the extension of the compensated path</p> <p>Contour damage</p> |
|---|---|

Radius compensation must be switched off after departure from the contour (G40).



Illustration

- programmed path
- - - traversed cutter center path

Common starting and end point

For a **common starting and end point**, select point ③ on the bisecting line of the angle between the first and last contour element.

Contour Approach and Departure

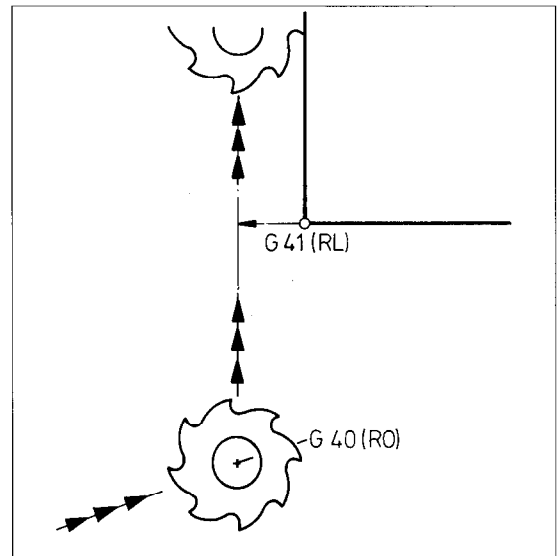
Starting and end position



Approach

The starting position must be programmed without radius compensation, i.e. with G40.

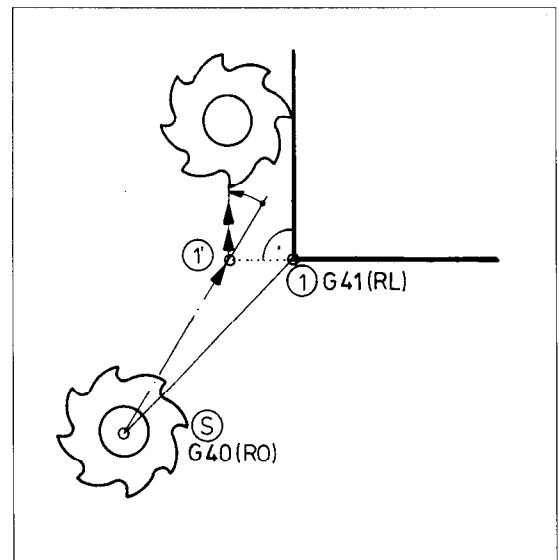
The control guides the tool in a straight line from the uncompensated position ③ to the compensated position ① of contour point ①. The tool center is then located perpendicular to the start of the first radius-compensated contour element.



Departure

At a transition from G41/G42 to G40, the control positions the tool center in the last radius compensated block (G41) perpendicular to the end of the last contour section.

Then the next uncompensated position is approached with G40.

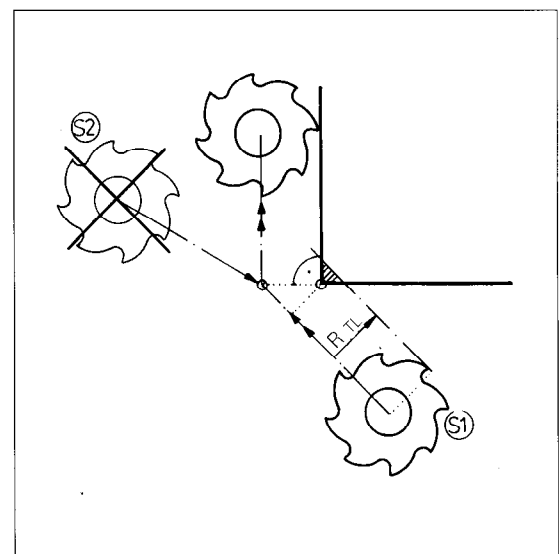


Approaching from an unsuitable position

If radius compensation is begun from S1, the tool will damage the contour at the first contour point if no extra measures are taken!

Departure

The same applies when departing from the contour.



Contour Approach and Departure on a circle with radius R: G26/G27



Approach and departure on an arc G26/G27

The TNC enables you to automatically approach and depart from contours on a circular path.

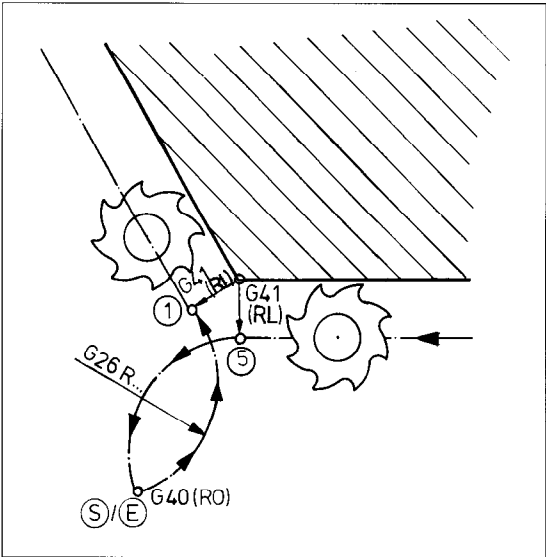
Begin programming with the G26 or G27 key.

Approach

The tool moves from the starting position ⑤ initially on a straight line and then on a tangentially connected arc to the programmed contour. The starting point can be selected as desired, and is approached without radius compensation (with G40).

The straight line positioning block to contour point ① must contain radius compensation (G41 or G42).

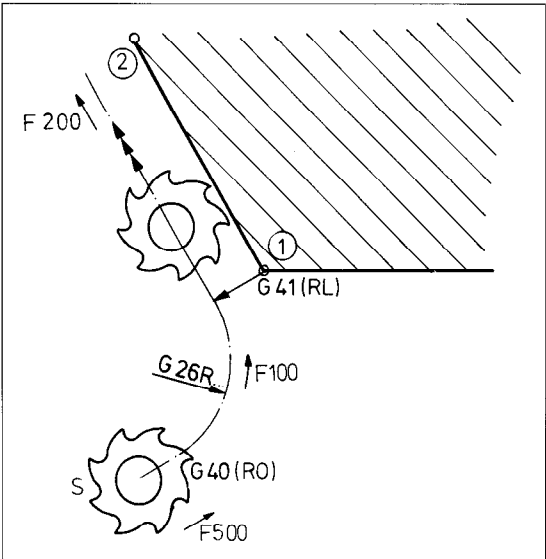
Then program a G26 block.



Departure

The tool moves from the last contour point ⑤ on a tangentially connecting arc and then on a tangentially connecting straight line to the end position ⑥ if a block with G27 is programmed between ⑤ and ⑥!

The positioning block for ⑥ should not contain radius compensation (i.e. G40).



Approach arc/ departure arc

The radius R can be substantially less than the tool radius. It must be small enough to fit between ⑤ and ① or ⑤ and ⑥.

Feed rate

A feed rate exclusively for the approach and departure arc can be programmed separately in the G26/G27 block.

Program scheme

G00 G40 X ₅ Y ₅ Z ₅	⋮
G01 G41 X ₁ Y ₁ F500	G41 X ₅ Y ₅ F200
G26 R2.5 F100	G27 R2.5 F100
X ₂ Y ₂ F500	G40 X _E Y _E F500
⋮	G00 Z+200

Notes

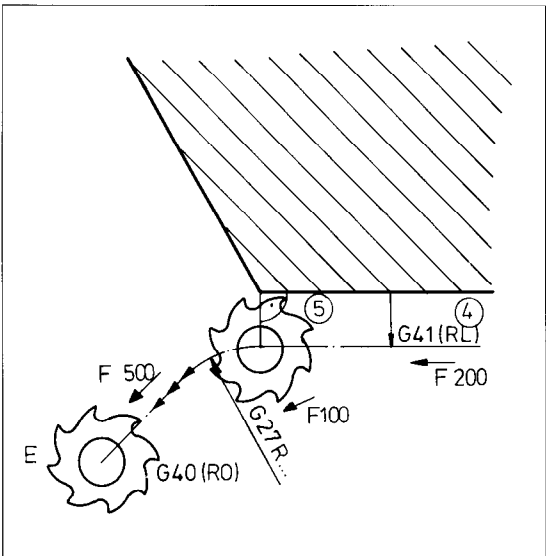
A positioning block containing both coordinates of the machining plane must be programmed before and after the G26/G27 block.

Approach on an arc:

Program a G26 block after the first radius compensated position (G41/G42).

Departure on an arc:

Program a G27 block after the last radius compensated position (G41/G42), or before the first uncompensated position following machining.



Predetermined M Functions

Constant contour speed: M90



**Standard practice:
automatic deceleration
at corners**

For angular transitions such as internal corners and contours with G40, the axes are stopped briefly because an abrupt change of direction is not mechanically possible.

This protects the machine and results in sharp definition of corners.

For some tasks it is advantageous not to stop at corners.

Example:
The contour of a free-form surface produced with a large number of short linear movements. Here it is desirable to smooth the corners.

M90

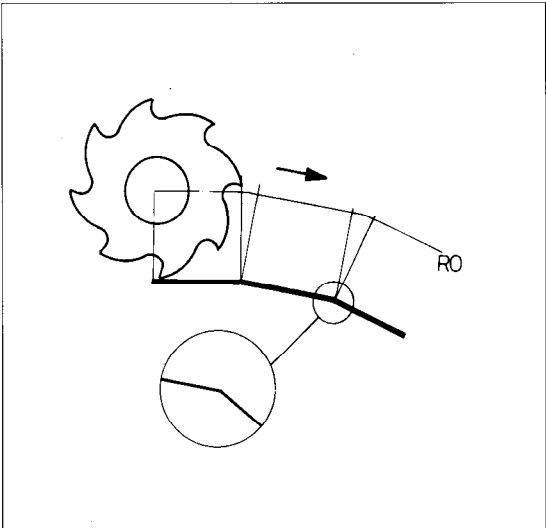
The corners are **smoothed** if **M90** is programmed in every block. The workpiece is smoother and can be machined faster. M90 prevents stoppage of the axes blockwise for G40 or internal corners.

Drawbacks

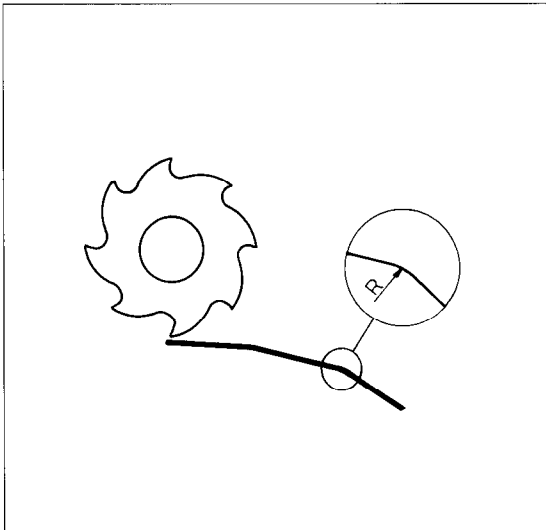
Greater strain on the machine at sharper changes of direction, until safety limit is reached (specified by the machine manufacturer).

Note

The exact execution depends on the machine parameters. Contact the machine manufacturer for more information.



Without M90



With M90

Predetermined M Functions

Small contour steps: M97



If there is a step in the contour which is **smaller** than the tool radius, the standard transition arc would cause contour damage. The control therefore issues the error message "TOOL RADIUS TOO LARGE" and does not execute the corresponding positioning block.

M97

M97 prevents insertion of the transition arc. The control then determines a contour intersection © as at inside corners and guides the tool over this point. The contour is not damaged.

However, machining is then incomplete and the corner may have to be reworked. A smaller tool may help.

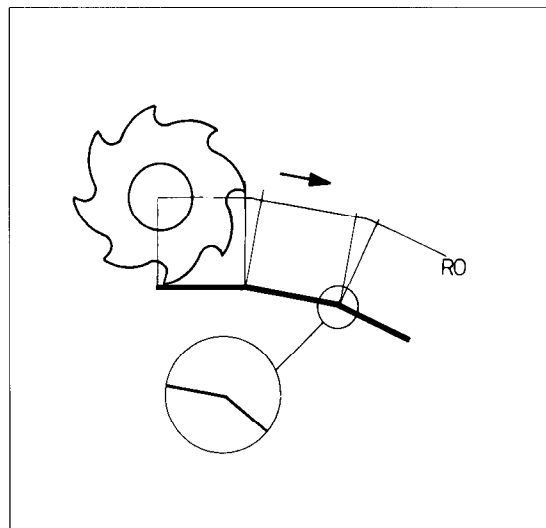
M97 is effective blockwise and must be programmed in the block containing the outside corner point.

Example

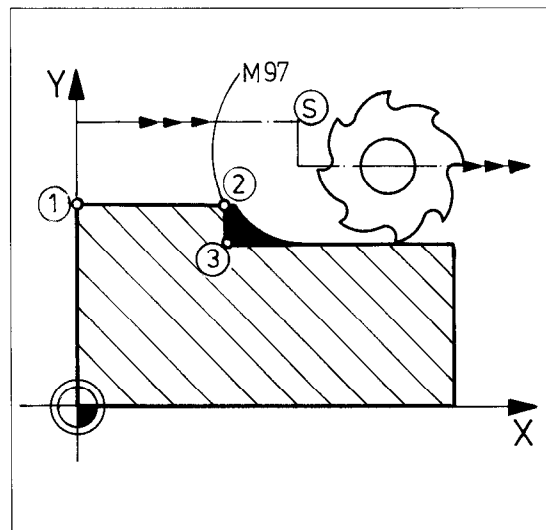
```
G99 T1 L0 R10 *
T1 G17 S100 *
```

```
G01 G41 X+10 Y+30 F200 M3 *
X+40 Y+30 M97 *
X+40 Y+28 *
X+80 Y+28 *
X+80 Y+30 M97 *
X+100 Y+30 *
```

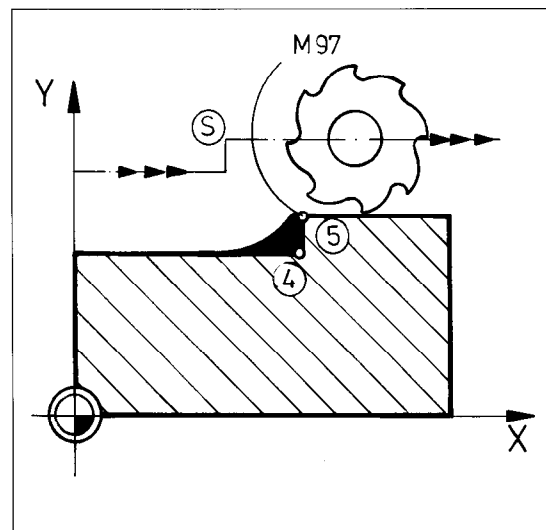
- ①
- ②
- ③
- ④
- ⑤



Without M97



With M97



With M97

Predetermined M Functions

Terminating compensation: M98



Standard inside corner compensation

On inside corners in a continuously radius-compensated contour, the tool moves only to the intersection of the equidistants (see top figure). The work cannot be completely machined at positions ③ and ④.

M98

The middle figure shows two independent work-pieces. Positions ③ and ④ are not connected. The tool must therefore be guided to positions ⑤ and ⑥.

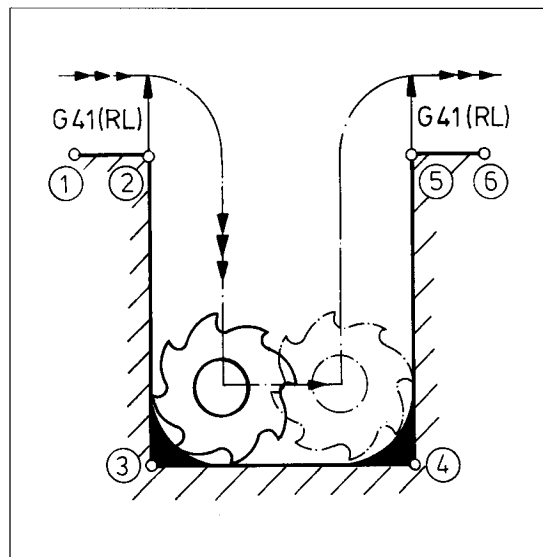
If you program a position with M98, the path offset remains valid until the end of this element and is ended there **for this block**.

No intersection is computed and no transition arc is generated for the end position, so the tool is always moved to a point perpendicular to the contour at its end point.

The previous compensation is reactivated automatically in the following block ④.

Position ④ is approached perpendicularly to ④.

The contour is thus completely machined at ③ and ④.

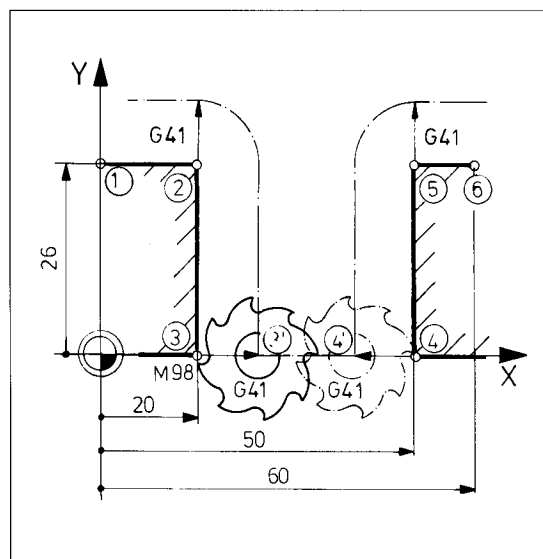


Example

```
G01 G41 X0 Y26 F100 * ①
X+20 Y+26 * ②
X+20 Y+0 M98 * ③
X+50 Y+0 * ④
X+50 Y+26 * ⑤
X+60 Y+26 * ⑥
```

Multipass milling with M98

Multipass milling with infeeds in Z.



Example

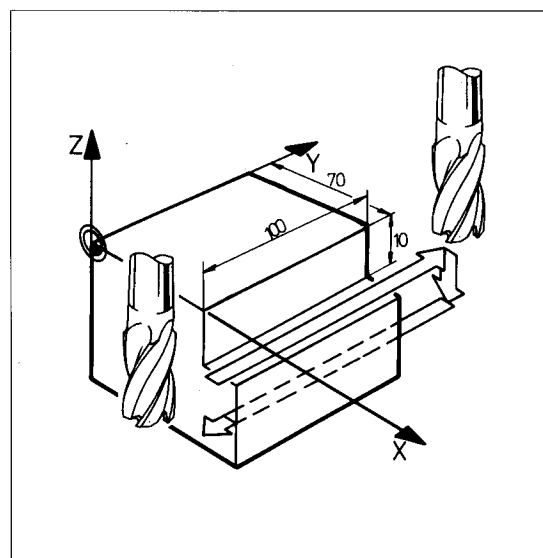
```
G30 G17 X+0 Y+0 Z-40 *
G31 G90 X+100 Y+100 Z+0 *
G99 T1 L+0 R+5 *
T1 G17 S200 *

G00 G90 Z+50 * Pre-positioning
G42 X+70 Y-10 M03 *

Z-10 * Tool-axis infeed

G01 Y+110 F200 M98 * Mill one pass
G00 Z-20 * Second tool-axis infeed
G01 G41 Y+110 F200 * Pre-positioning
Y-10 M98 * Mill second pass

G00 Z+50 * Retract
```



Predetermined M Functions

Programming machine-based coordinates: M91/M92



**Standard
behaviour**

Coordinates in positioning blocks are based on the workpiece datum.

Scale datum

The position of the scale datum is determined by the reference marks. If the scale has only one reference mark, then the reference mark is the scale datum. If the scale has several – distance-coded – reference marks, then the leftmost reference mark is scale datum (beginning of the measuring length). With the TNC 360 the scale datum point is the same as the machine datum point.

**Machine datum:
M91**

The machine datum is required for the following:

- Setting the traverse range limits (software limit switch)
- Traversing to machine-based positions (such as tool change positions)
- Setting the workpiece datum

If the coordinates in positioning blocks are based on the machine datum, enter M91 in these blocks.

Coordinates are displayed referenced to the machine datum with the coordinate display REF.

**Additional
machine
reference
point:
M92**

The machine builder can also define an additional machine-based reference point.

The machine builder enters the distance from the machine datum to this additional machine reference point.

If the coordinates in positioning blocks are based on this additional machine reference point, enter M92 in these blocks.

Program Jumps

Overview



Jumping within a program

The following jumps can be made within a program:

- **Program section repeat**
- **Subprogram call**
- **Conditional jump**
- **Unconditional jump**

Nesting:

A further program section repeat or subprogram can be called up from within a program section repeat or subprogram.

Maximum nesting depth: 8 levels

Examples:

L 4,3 *

L 7,0 *

D11 P01+Q5 P02+0 P03 12 *

D09 P01+0 P02+0 P03 8 *

Jumping to another program

You can jump from one part program into any other program which is in the control's memory or on an external data storage medium. The jump into another program is programmed with a

- **Program call** with "PGM CALL" or
- **Cycle G79**, if another cycle was previously defined with G39 as a callable cycle.

Nesting:

You can call further programs from a called program.

Maximum nesting depth: 4 levels

Examples:

%3 *

G39 P01 3 *

G79 *

.
.
.
.

G01 X+50 M99 *

Jumps Within a Program

Program labels: G98



Labels

Labels (program markers) can be set during programming to mark the beginning of a subprogram or program section repeat.

These labels can be jumped to during program run (e.g. to execute the appropriate subprogram).

Setting a label G98

A label is set with the G98. The label numbers 1 to 254 can be set only once in a program.

Label 0

Label number 0 always marks the end of a subprogram (see "Subprogram") and is therefore the return jump marker. It can thus occur more than once in a program.

Do not call label 0!

Calling a label number



With the "L" key you can:

- call subprograms
- create program section repeats.

Label numbers (1 to 254) can be called as often as desired.

Do not call label 0!

```
%1 G71 *
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T1 L+0 R+3 *
N40 T1 G17 S500 *
N50 G83 P01 -2 P02 -20
      P03 -6 P04 0 P05 120 *
N60 G00 G90 Z+50 M06 *
N70 G40 X+10 Y+20 M03 *
N80 Z+2 *
N90 L1.0 *
N100 X+20 Y+50 *
N110 L1.0 *
N120 X+10 Y+80 *
N130 L1.0 *
N140 G00 Z+50 M02 *
```

```
N150 G98 L1 *
N160 G79 *
N170 G98 L2 *
N180 G00 G91 X+10 M99 *
N190 L2.5 *
N200 G90 *
N210 G98 L0 *
N9999 %1 G71 *
```

Pecking cycle
Refer to "Fixed cycles" for explanation

Program section repeats

For program section repeats, enter the required number of repetitions (e.g. L2,5).

Subprograms

For subprogram calls, enter 0 as the number of repetitions (e.g. L1,0), or simply conclude with "End □".

Error messages

JUMP TO LABEL 0 NOT PERMITTED

This jump (L0) is not allowed.

LABEL NUMBER ALLOCATED

Each label number – except L 0 – can be allocated (set) **only once** in a given program.

Jumps Within a Program

Program section repeats



Program section repeats

Once a program section has been executed, it can be executed again immediately. This is called a program loop or program section repeat.

G98

A label number marks the beginning of the program section which is to be repeated.



with number

The end of the program section to be repeated is designated by a call LBL CALL with the number of repetitions REP.

A program section can be repeated up to 65534 times.

Jump direction

A called program section repeat is always executed completely, i.e. up to L.

A program jump is therefore only meaningful if it is a **return jump**.

Program run

The control executes the main program (along with the associated program section) until the label number is called. Then the return jump is carried out to the called program label and the program section is repeated.

The number of remaining repetitions on the display is reduced by 1: L 2/5.

After another return jump, the program section is repeated a second time.

When all programmed repetitions have been performed (display: L 2/0), the main program is resumed.

The total number of times a program section is executed is always one more than the programmed number of repeats.

```
N22 G98 L2 *  
N23 G00 G91 X+100 M99 *  
N24 L2,5 *  
.  
.  
.
```

```
N22 G98 L2 *
```

```
N23 G00 G91 X+10 M99 *
```

```
N24 L2,5 *  
.  
.  
.
```

Error message

EXCESSIVE SUBPROGRAMMING

You programmed a jump incorrectly:

You failed to enter the repetition value. The program section is treated as a subprogram without a correct ending (G98 L0): the label number is called eight times. During program run or a test run the error message appears on the screen after the eighth repetition.

Jumps Within a Program

Program section repeats



Setting the program label

Example:

Program label 1 is set.

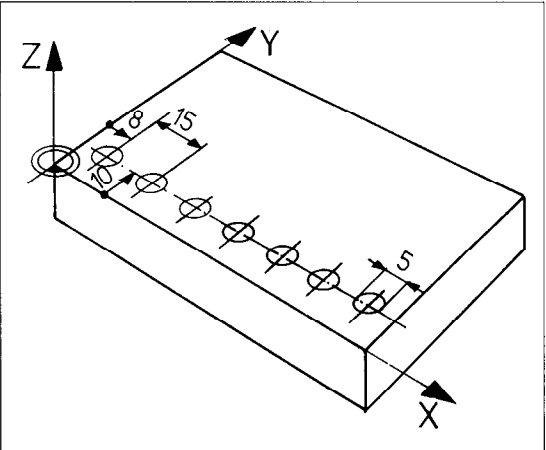
Repeating a program section after a label

6 repetitions from G98 L1.
 The program section between G98 L1 and L1,6 is executed a total of 7 times.

Example bolt-hole row

The illustrated bolt-hole row with 7 identical bores is to be drilled with a program section repeat.

The tool is pre-positioned (offset to the left by the bore center distance) before starting the repeat to simplify programming.



Program

G99 T1 L+0 R2.5 *
 T1 G17 S200 *
 G00 G40 G90 X-7 Y+10 Z+2 M03 *

Tool definition
 Tool call
 Pre-positioning

G98 L1*
 G91 X+15 *

 G01 G90 Z-10 F100 *
 G00 Z+2 *
 L1,6 *

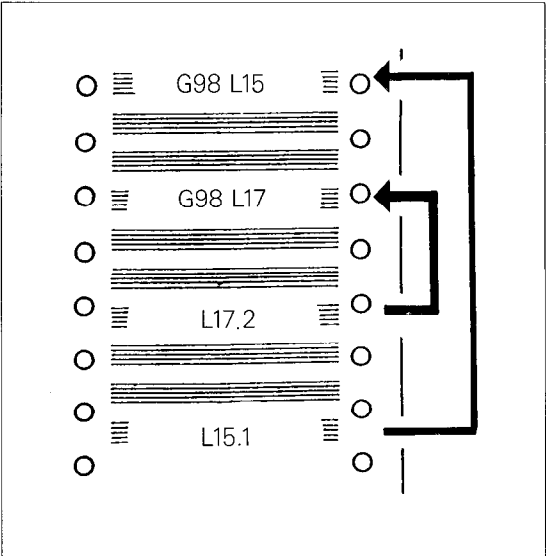
Start of the program section repeat
 Incremental distance between the bores, rapid traverse
 Absolute drilling depth, drilling feed rate
 Absolute retraction height, rapid traverse
 Call for repeats

Nesting of repetitions

The main program is executed until the jump to G98 L17 (L17,2).
 The program section between G98 L17 and L17,2 is repeated twice.

The control then resumes the main program run until the jump to G98 L15 (L15,1).

The program section up to L15,1 is repeated once and the nested program section also two more times. Then the program run is continued.



Jumping Within a Program

Subprograms



Subprograms	If a program section occurs several times in the same program, it can be designated as a subprogram and called whenever required. This speeds up programming.
Start of subprogram	The start of the subprogram is marked with a label number (can be any number).
End of subprogram	The end of the subprogram is always marked by label 0 . The different subprograms are then called in the main program as often as wanted and in any sequence.

```
.  
. .  
. .  
N14 L1,0 *  
N15 G01 X+20 Y+50 *  
N16 L1,0 *  
N17 X+10 Y+80 *  
N18 L1,0 *  
N19 G00 G40 Z+50 M02 *  
  
N20 G98 L1 *  
N21 G79 *  
N22 G98 L2 *  
N23 G00 G91 X+10 M99 *  
N24 L2,5 *  
N25 G98 L0 *
```

No repetitions	For a subprogram call with the “L” key, the block is concluded after the label number with “END □”. A subprogram can be called at any point in the main program (but not from within the same subprogram).
-----------------------	--

Program run	The control executes the main program until the subprogram call ①. A jump to the called program label ② is then performed. Subprogram 1 is executed until G98 L0 (③) (end of subprogram). Then the return jump to the main program follows. The main program is resumed with the block ④ following the subprogram call.
--------------------	--

```
.  
. .  
. .  
① L1,0 *  
④ G01 X ... Y ...  
. .  
. .  
. .  
M02 *  
  
② G98 L1 *  
. .  
. .  
③ G98 L0 *
```

Subprograms should be placed after the main program (behind M2 or M30) for the sake of clarity. If a subprogram is placed within the main program, it is also executed once during program run without being called.

Error messages	If a subprogram call is programmed incorrectly (e.g. an end of subprogram lacks G98 L0), the error message EXCESSIVE SUBPROGRAMMING appears.
-----------------------	---

Jumps Within a Program

Nesting subprograms



Nesting subprograms

The main program is executed until the jump command L17,0 is reached.

The subprogram beginning with G98 L17 is subsequently executed until the next call L20, which is then run until L53,0.

The lowest nested subprogram 53 is run through until its end (G98, L0).

At the end (G98 L0) of the last subprogram (53), return jumps are made to the preceding subprograms (20 and 17), until the main program is finally reached.

The main program is then taken up again at the point immediately following the call L17,0.

A subprogram call is considered executed when the first G98 L0 is reached!

```
%12 G71 *  
.  
.  
L17,0 *  
.  
.  
M02 *
```

```
G98 L17 *  
.  
.  
L20,0 *  
.  
.  
G98 L0 *
```

```
G98 L20 *  
.  
.  
L53,0 *  
.  
.  
G98 L0 *
```

```
G98 L53 *  
.  
.  
G98 L0 *
```

```
N9999 %12 G71 *
```

Repeating subprograms

You can execute subprograms repeatedly with the nesting technique:

Subprogram 50 is called in a program section repeat. This subprogram call is the only block in the program section repeat.

Remember: the subprogram will be executed one more time than the programmed number of repeats.

```
.  
.  
.  
G98 L5 *  
L50,0 *  
L5,9 *  
.  
.  
.  
M2 *
```

Error message

If too many nesting levels were programmed, the error message EXCESSIVE SUBPROGRAMMING appears.

```
G98 L50 *  
.  
.  
.  
G98 L0 *
```

Jumps Within a Program

Example: Hole pattern with several tools

Task This task is similar to the example of the "group of four bores at three different positions" (see chapter "Jumps Within a Program", section "Sub-program") except that here three different tools and machining processes are to be used.

Note You will find an explanation of the pecking and tapping cycles in the chapter "Fixed cycles".

```
%183 G71 *
N10 G30 G17 X+0 Y+0 Z-20 *
N20 G31 G90 X+110 Y+100 Z+0 *
N30 G99 T25 L+0 R+2.5 *
N40 G99 T30 L+0 R+3 *
N50 G99 T35 L+0 R+3.5 *
```

Countersink

```
N60 G83 P01 -2 P02 -3 P03 -3
      P04 0 P05 100 *
N70 T35 G17 S1000 *
```

```
N80 G00 G90 Z+50 M06 *
N90 L1,0 *
```

Pecking

```
N100 G83 P01 -2 P02 -25 P03 -6
      P04 0 P05 50 *
N110 T25 G17 S2000 *
N120 G00 Z+50 M06 *
N130 L1,0 *
```

Tapping

```
N140 G84 P01 -2 P02 -15
      P03 0 P04 100 *
N150 T30 G17 S250 *
N160 G00 Z+50 M06 *
N170 L1,0 *
N180 G00 G40 Z+50 M02 *
```

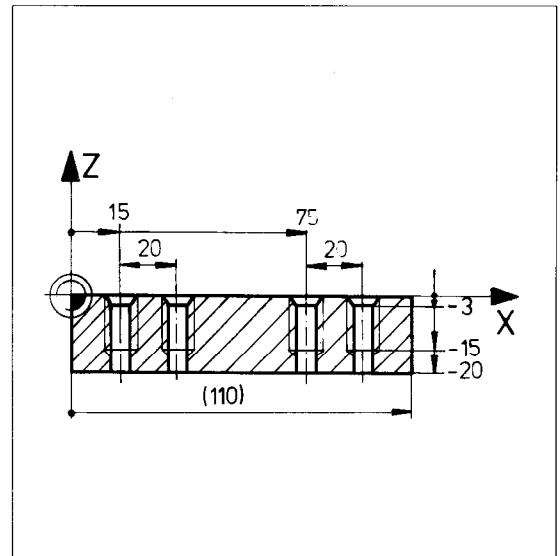
Subprogram 1

```
N190 G98 L1 *
N200 G40 X+15 Y+10 M03 *
N210 Z+2 *
N220 L2,0 *
N230 X+45 Y+60 *
N240 L2,0 *
N250 X+75 Y+10 *
N260 L2,0 *
N270 G98 L0 *
```

Subprogram 2

```
N280 G98 L2 *
N290 G79 *
N300 G91 X+20 M99 *
N310 Y+20 M99 *
N320 X-20 M99 *
N330 G00 G90 *
N340 G98 L0 *
```

```
N9999 %183 G71 *
```



Tool change
Call: subprogram 1

Tool change

Tool change

Retract spindle axis, jump to start of program

Approach hole pattern ①
Move to setup clearance
Call: subprogram 2
Approach hole pattern ②

Approach hole pattern ③

Cycle call (countersink, peck drill, tap)
M99 = blockwise cycle call

Jumping Within a Program

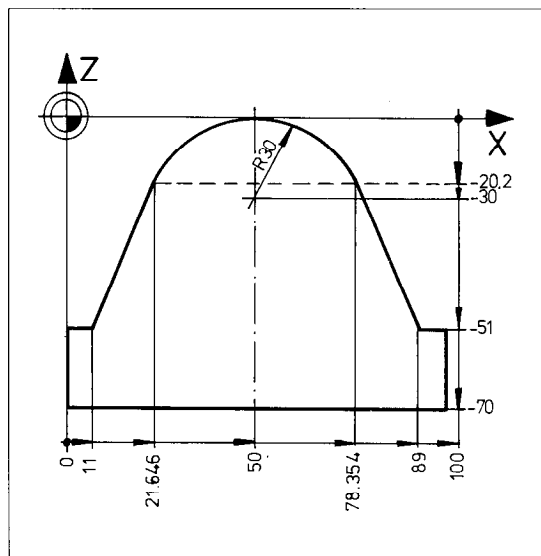
Example: Horizontal geometric form

Task

The adjacent geometric contour is to be machined from a cuboid with an end mill which is to be advanced stepwise in the Y direction by a program section repeat.

The contour is divided into two halves along the line of symmetry to simplify the working procedure. The contour is to be machined upwards.

In addition to the adjacent dimensions, the cuboid length is specified with:
Y = 100 mm.



Program procedure

The adjacent figure schematically shows the cutter center path and the associated program blocks. The entire contour is divided into a "left" and "right" half and is machined in the two program section repeats.

The program runs **without radius compensation**, i.e. the cutter center path is programmed. To obtain the desired contour, the tool radius must be **subtracted** on the **left** side and **added** on the **right** side (all X coordinates).

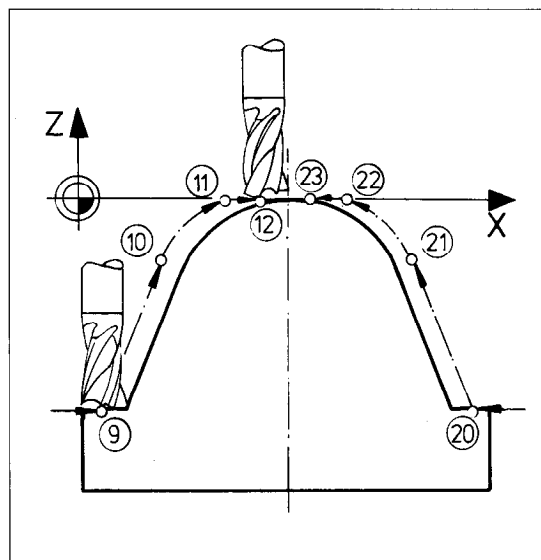
```
%90007685 G71 *
N10 G30 G17 X+0 Y+0 Z-70 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T1 L+0 R+10 *
N40 T1 G17 S1000 *
N50 G00 G90 Z+20 M06 *
N60 G40 X-20 Y-1 M03 *
```

Program section repeat 1

```
N70 G98 L1 *
N80 Z-51 *
N90 G01 X+1 F100 * (9)
N100 X+11.646 Z-20.2 * (10)
N110 G06 X+40 Z+0 * (11)
N120 G01 X+41 * (12)
N130 G00 Z+10 *
N140 X-20 G91 Y+2.5 *
N150 G00 G90 *
N160 L1,40 *
```

Program section repeat 2

```
N170 G00 Z+20 *
N180 X+120 Y-1 *
N190 G98 L2 *
N200 Z-51 *
N210 G01 X+99 F100 * (20)
N220 X+88.354 Z-20.2 * (21)
N230 G06 X+60 Z+0 * (22)
N240 G01 X+59 * (23)
N250 G00 Z+10 *
N260 X+120 G91 Y+2.5 *
N270 G00 G90 *
N280 L2,40 *
N290 Z+20 M02 *
N9999 %90007685 G71 *
```



Approach starting point for "left side"

Infeed in Y axis

Program section is executed 41 times

Retract spindle axis

Approach starting point for "right side"

Infeed in Y axis

Program section is executed 41 times

Retract spindle axis, jump to start of program



Jumping to another main program

You can call another program which is stored in the control from any machining program. This allows you to create your own fixed cycles with parametric programming. Program the call with a “%” key.

Calling criteria

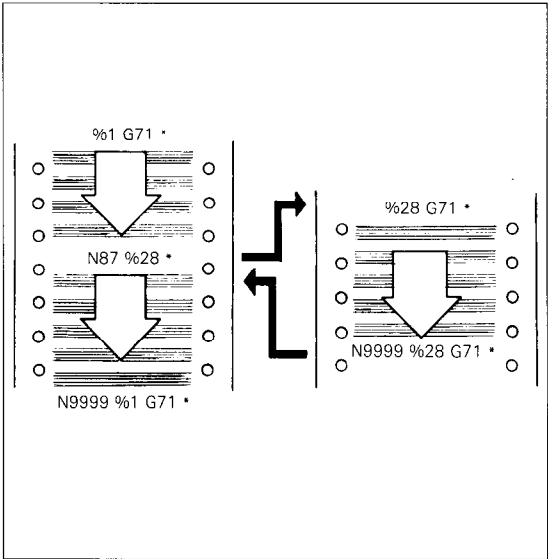
The program to be called cannot contain M02 or M30. In the called program, do not program a jump back to the original program (creates an endless loop). Only one BLK FORM can exist. Tool numbers may be assigned only once.

Process

The control executes main program 1 up to the program call %28. Then a jump is made to main program 28.

Main program 28 is executed from beginning to end.

Then a return jump is made to main program 1. Main program 1 is resumed with the block following the program call.



Example 1



% 10 *

Call with a separate program line

Example 2

The program to be called can also be specified with a cycle definition. The call then functions like a fixed cycle.



G39 P01 12 *

Call e.g. via M99
(see Cycle G39)

Conditional jumps

A label call can be made dependent on a mathematical condition (see “Parametric Programming, Overview, Basic functions”).

Standard Cycles

Introduction, Overview

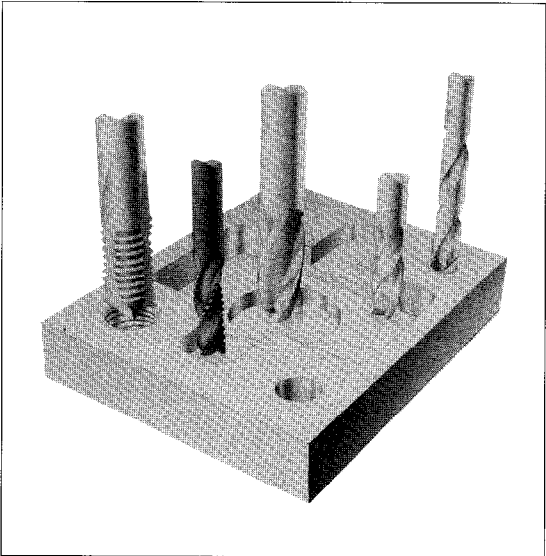


- Standard cycles**

To facilitate programming, frequently recurring machining sequences (drilling and milling jobs) and certain coordinate transformations are pre-programmed as standard cycles.
- Machine builder cycles**

The machine manufacturer can also store his own programs as cycles in the control.

These cycles can be called under the cycle numbers 68 to 99. Contact the machine manufacturer for more information.



- Selecting a cycle**
- After selecting the appropriate G-function and pressing the "ENT" key, data for the cycles shown to the right can be entered and also any programmed user cycles can be selected.

Fixed cycles	G	Cycle	Effective after call	Effective immediately
	83	Pecking	●	
	84	Tapping	●	
	74	Slot milling	●	
	75/76	Rectangular pocket	●	
	77/78	Circular pocket	●	
	79	Program call	●	
	73	Contour geometry		●
	56	Pilot drilling	●	
	57	Rough-out	●	
	58/59	Contour milling	●	

- Calling a fixed cycle**
- Cycles must be called after moving the tool to the appropriate position – only then will the last defined cycle be executed.

There are three ways to call a cycle:

G79

M99

M89

- With the cycle call function G79
- Via the miscellaneous function M99. G79 and M99 are only effective blockwise and must therefore be reprogrammed for every execution.
- Via the miscellaneous function M89 (depending on machine parameters).

M89 is effective modally, i.e. the last programmed cycle is called at every subsequent positioning block. M89 is cancelled or cleared by M99, G79 or by newly defining a fixed cycle.

- Coordinate transformations**
- Coordinate transformations and the dwell time are effective immediately and remain effective until changed.

Coordinate transformations	G	Cycle	Effective after call	Effective immediately
	54	Datum shift		●
	28	Mirror image		●
	73	Rotating the coordinate system		●
	72	Scaling		●
	04	Dwell time		●
	39	Program call		●
	36	Spindle orientation		●

Fixed Cycles

Preparatory measures



Prerequisites	The following must be programmed before a cycle call (e.g. M99). • Tool call: to specify the spindle axis and the spindle speed • Positioning block to the starting position
Dimensions	In the cycle definition, dimensions for the tool axes are to be entered incrementally, referenced to the tool position at cycle call. All infeeds must have the same sign (usually negative).
Entering values	Enter all values as requested and confirm entry with "ENT". You must respond to every dialog query by entering a value! Conclude entry with "END ☐".

T1 *

G0 G90 X ... Y ... M3 *

G83 *

Z ... M99 *

Fixed Cycles

Pecking: G83



Function

A hole is drilled with multiple infeeds, each followed by a complete retraction.

Input data

Infeed value signs:

- - for negative working direction
- + for positive working direction

All infeeds must have the same sign.

Setup clearance A: distance between tool tip (starting position) and workpiece surface.

Total hole depth B: distance between the workpiece surface and the bottom of the hole (tip of the drill taper).

Pecking depth C: the infeed per cut.

Dwell time: the time the tool remains at the bottom of the bore hole for chip breaking.

Feed rate F: traversing speed of the tool during infeed.

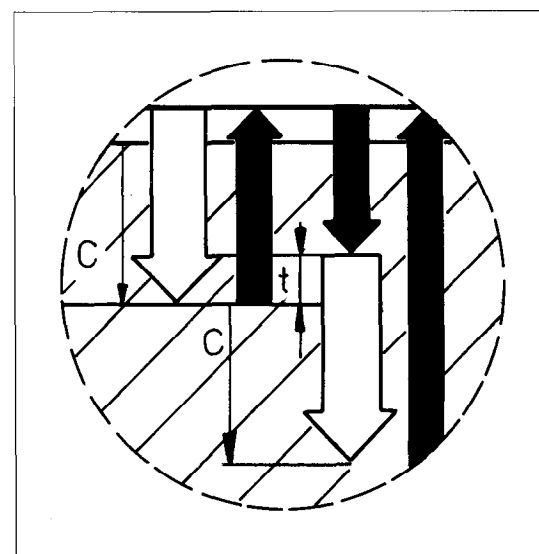
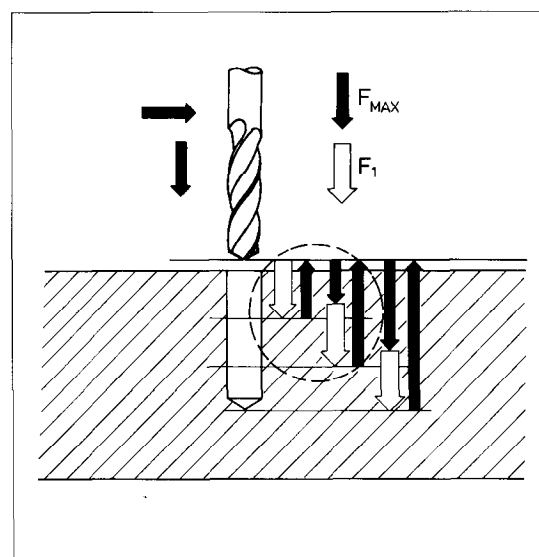
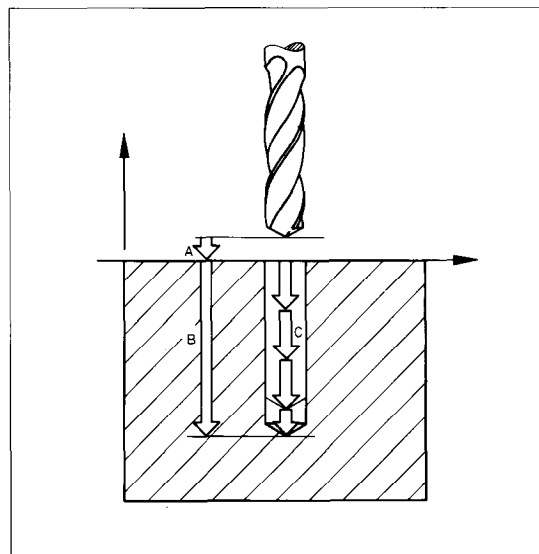
Process

- The tool must be positioned to the setup clearance with a **separate block**, before the cycle call.
- The tool drills from the starting position to the first pecking depth at the programmed feed rate.
- After reaching the first pecking depth the tool is retracted in rapid traverse to the starting position and advanced again to the first pecking depth, minus the advanced stop distance t .
- The tool then advances by another infeed at the programmed feed rate, returns again to the starting position etc.
- Drilling and retraction are performed alternately until the programmed total hole depth is reached.
- After the dwell time at the hole bottom, the tool is retracted to the starting position in rapid traverse.

Advanced stop distance

The advanced stop distance t is automatically computed by the control:

- For a total hole depth up to 30 mm:
 $t = 0.6 \text{ mm}$;
- For a total hole depth over 30 mm:
 $t = \text{total hole depth}/50$, whereby the maximum advanced stop distance is limited to $t_{\max} = 7 \text{ mm}$.



Fixed Cycles

Pecking: G83



Defining the cycle

Operating mode



SET UP CLEARANCE ?

- ☐ Specify setup clearance
- ☒ Enter the sign correctly (normally positive)
- ☒ Confirm entry

TOTAL HOLE DEPTH ?

- ☐ Specify hole depth
- ☒ Enter the sign correctly (normally negative)
- ☒ Confirm entry

PECKING DEPTH ?

- ☐ Specify pecking depth
- ☒ Enter the sign correctly (normally negative)
- ☒ Confirm entry

DWELL TIME IN SECS. ?

- ☐ Enter the dwell time at the bottom of the hole (zero for no dwell time)
- ☒ Confirm entry

FEED RATE ? F =

- ☐ Enter the feed rate for pecking
- ☒ Confirm entry

The signs for setup clearance, total hole depth and pecking depth are all the same (normally negative)!

Fixed Cycles

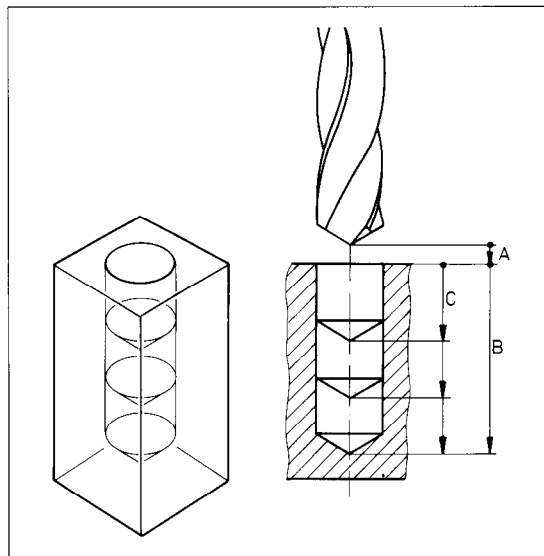
Pecking: G83



Remarks

- The total hole depth can be programmed equal to the pecking depth. The tool then traverses in **one** work step to the programmed depth (e.g. for centering).
- The total depth need not be a multiple of the pecking depth. In the last work step, the tool will only be advanced the remaining distance to the programmed hole depth.
- If the specified pecking depth is greater than the total hole depth, drilling is only performed to the programmed total hole depth.

The above also applies to other fixed cycles.



Example

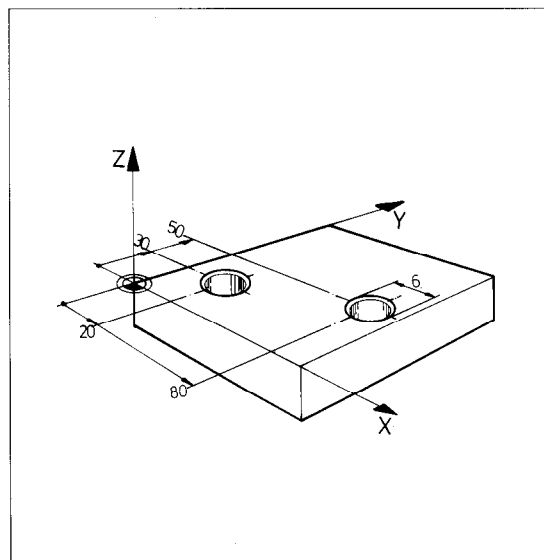
Drill 2 holes (depth 20 mm) with the standard pecking cycle.

G99 T1 L+0 R3 *
T1 G17 S200 *

Tool definition
and call

G83 P01 -2
P02 -20
P03 -10
P04 2
P05 80 *

Setup clearance
Total depth
Pecking depth
Dwell time
Feed rate



G00 G40 X+20 Y+30 M03 *

Pilot positioning, spindle on

Z+2 M99 *

1st hole, cycle call

X+80 Y+50 M99 *

2nd hole, cycle call

Fixed Cycles

Tapping with floating tap holder: G84



Function

The thread is cut in **one** operation.

A **floating tap holder** is required for tapping. It must compensate for the tolerances between the feed rate, speed and the tool geometry as well as spindle run out after the position is reached.

Spindle speed override is inactive after a cycle call; the feed rate override is only active over a limited range (set by the machine manufacturer via machine parameters).

Input data

Setup clearance A: distance between tool tip (starting position) and workpiece surface (standard value: approx. 4 x thread pitch). The preceding sign depends on the working direction.

Total hole depth B (= thread length): distance between workpiece surface and end of thread. The signs for setup clearance and total hole depth are the same (usually negative).

Dwell time: enter either the time between reversing the direction of spindle rotation and retracting the tool, or 0. This time is machine-dependent.

Feed rate/ thread pitch

Feed rate F: traversing speed of the tool during tapping.

Determining the required feed rate:

$$F = S \times P$$

F: feed rate

S: spindle speed

P: thread pitch

The thread pitch is determined indirectly by the spindle speed specified in the tool call and the feed rate of the cycle (see index A, "General Information, Cutting Data").

Process

Once the tool has reached the total hole depth, the direction of spindle rotation is reversed within a time period set by machine parameters.

At the end of the programmed dwell time, the tool is retracted to the starting position. The spindle direction is reversed again in the retracted position.

Input

Same as for "Pecking".

Example

Tap an M6 hole with 0.75 mm pitch at 100 rpm.

G99 T1 L+0 R3 *
T1 G17 S100 *

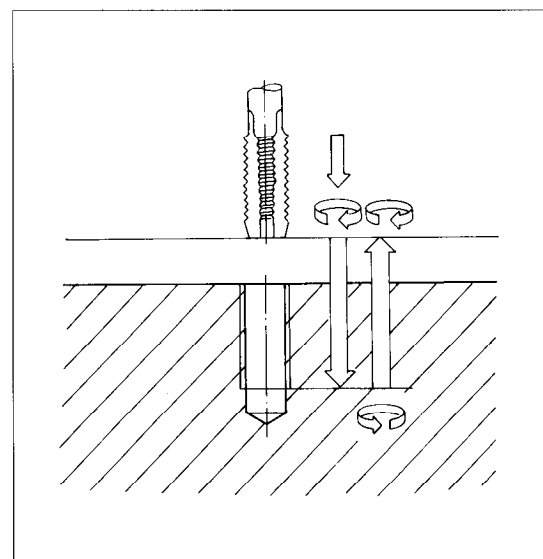
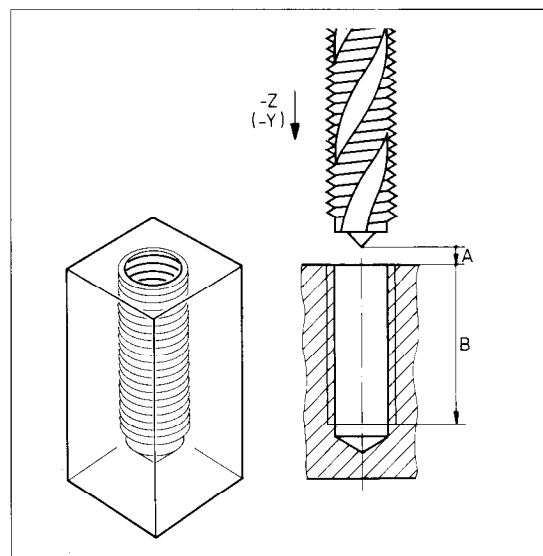
Tool definition
and call

G84 P01 -3
P02 -20
P03 0.4
P04 75 *

Setup clearance
Thread depth
Dwell time
Feed rate

G00 G40 X+50 Y+20 M03 *
Z+3 M99 *

Pilot positioning, spindle right
Cycle call



Fixed Cycles

Slot milling: G74



The cycle

The slot milling cycle is a combined roughing/finishing cycle.

The slot is parallel to one axis of the current coordinate system (rotation with cycle G73, if desired).

Tool required

The cycle requires a center-cut end mill (ISO 1641). The cutter diameter must be slightly smaller than the slot width.

Input data

Setup clearance A: distance between tool tip (starting position) and workpiece surface.

Milling depth B: (= slot depth): distance between work surface and bottom of slot.

Pecking depth C: penetrating distance of the tool into the workpiece.

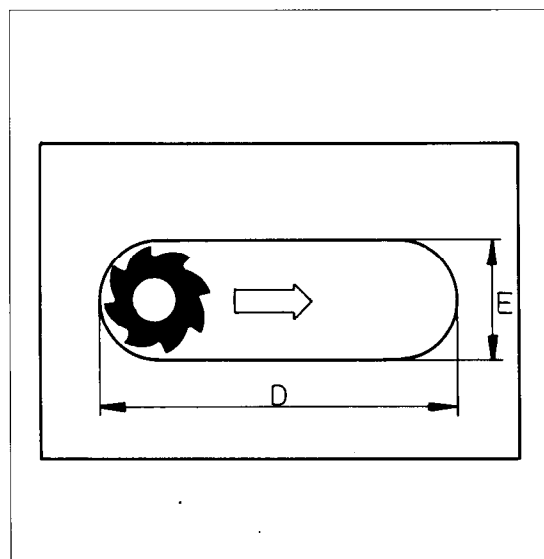
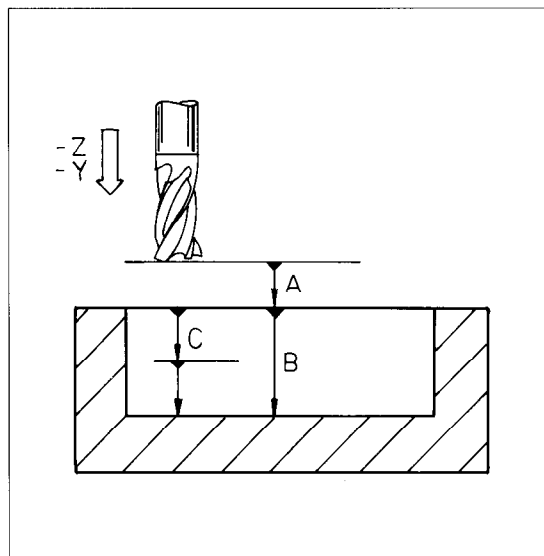
The signs for setup clearance, milling depth and pecking depth are all the same (usually negative).

Feed rate for pecking: traversing speed of the tool during penetration.

1st side length D: slot length (finished size). Sign depends on the first direction of cut parallel to the longitudinal axis of the slot.

2nd side length E: slot width, maximum 4 times the tool radius (finished size).

Feed rate: traversing speed of the tool in the machining plane.



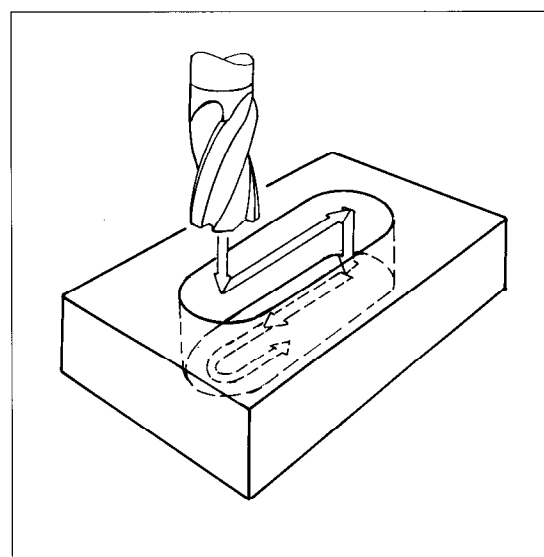
Roughing process

- The tool penetrates the work from the starting position.
- The slot is then milled in the longitudinal direction. After downfeed at the end of the slot, milling is in the opposite direction.
- The procedure is repeated until the programmed milling depth is reached.

Finishing process

The control advances the tool in a semicircle at the bottom of the slot by the remaining finishing cut and down-cut mills the contour (with M3). The tool is subsequently retracted in rapid traverse to the setup clearance.

If the number of infeeds was odd, the cutter returns along the slot at the setup clearance to the starting position in the main plane.



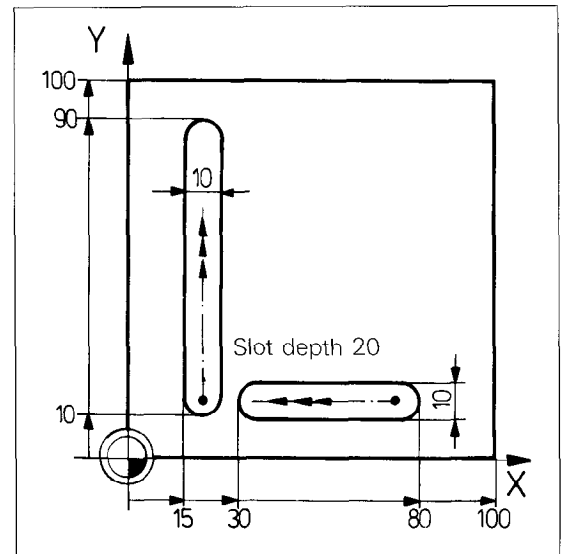
Fixed Cycles

Slot milling: G74



Example

A horizontal slot with length 50 mm and width 10 mm as well as a vertical slot with length 80 mm and width 10 mm are to be milled.



Cycle definition

```
N50 G74 P01 -2
      P02 -20
      P03 -5
      P04 80
      P05 X-50
      P06 Y+10
      P07 100 *
```

Definition of the **horizontal** slot

Setup clearance

Milling depth

Pecking depth

Feed rate for pecking

Length of slot and first **milling direction (-)**

Slot width

Feed rate

Starting position

```
N60 G00 G40 G90 X+76 Y+15 M3 *
```

Approach starting position **without compensation**, taking the tool radius into account in the longitudinal direction of the slot; spindle on
Pre-positioning in Z, cycle call

```
N70 Z+2 M99 *
```

Definition of the **vertical** slot

Setup clearance

Milling depth

Pecking depth

Feed rate for pecking

Slot length and first **milling direction (+)**

Slot width

Feed rate

```
N80 G74 P01 -2
      P02 -20
      P03 -5
      P04 80
      P05 Y+80
      P06 X+10
      P07 100 *
```

```
N90 X+20 Y+14 M99 *
```

```
N100 Z+50 M2 *
```

```
N9999 %5501 G71 *
```

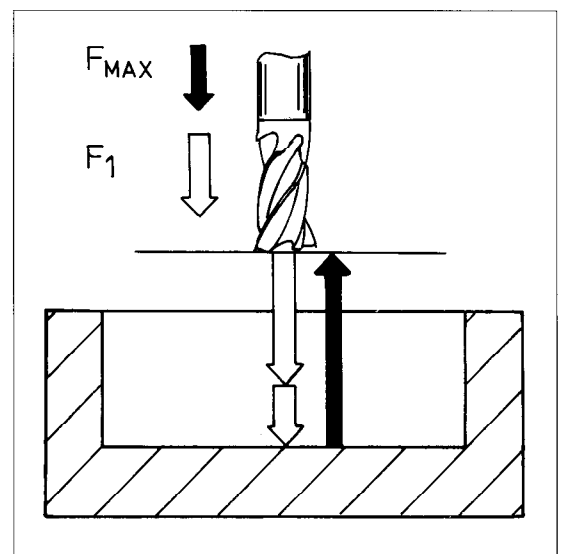
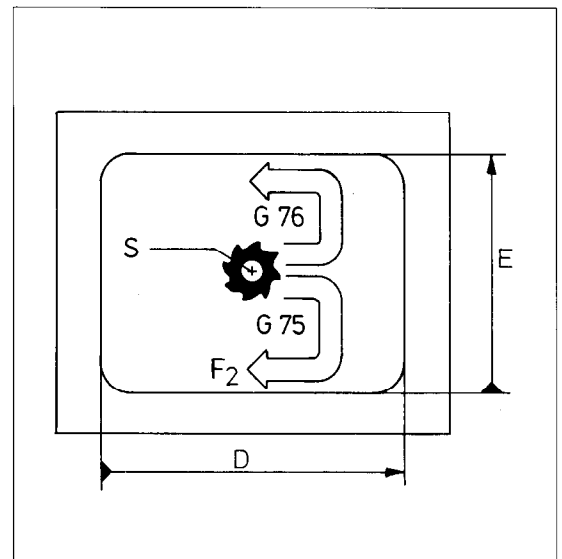
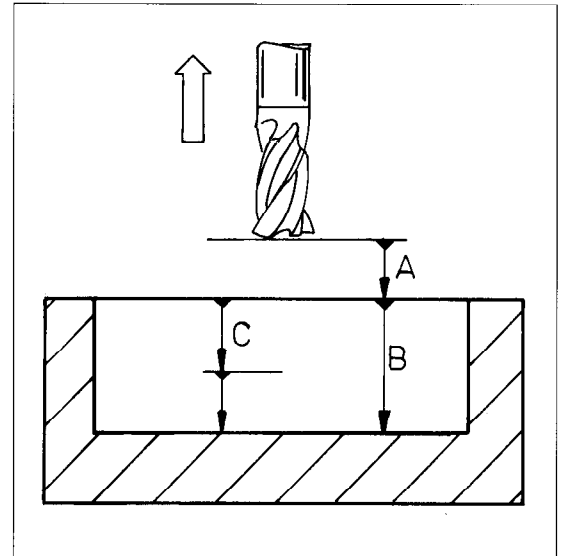
Approach starting position, cycle call
Retract in tool axis, end of program

Fixed Cycles

Rectangular pocket milling: G75/G76



The cycle	The rectangular pocket milling cycle is a roughing cycle.
Tool required	The cycle requires a center-cut end mill (ISO 1641), or pilot drilling at the pocket center. The tool determines the radius at the pocket corners. There is no circular movement in the pocket corners.
Position	The pocket sides are parallel to the coordinate system axes; the coordinate system may have to be rotated (see G73: Rotating the coordinate system).
Input data	Setup clearance A: distance between tool tip (starting position) and workpiece surface. Milling depth B (= pocket depth): distance between workpiece surface and bottom of pocket. Pecking depth C: distance by which the tool penetrates the workpiece. The signs for setup clearance, milling depth and pecking depth are all the same (usually negative). Feed rate for pecking F_1: traversing speed of the tool at penetration. 1st side length D: pocket length parallel to the first main axis of the machining plane. The sign is always positive. 2nd side length E: pocket width; the sign is always positive. Feed rate F_2: traversing speed of the tool in the machining plane. Direction of the milling path: Climb milling (down cut) G75: counterclockwise, down-cut milling with M3 Conventional milling (up cut) G76: clockwise, up-cut milling with M3
Starting position	The starting position S (pocket center) must be approached without radius compensation in a preceding positioning block.
Process	- The tool penetrates the work from the starting position (pocket center). - The cutter then follows the programmed path at feed rate F_2. The starting direction of the cutter is the positive axis direction of the longer side, i.e. if this longer side is parallel to the X axis, the cutter starts in the positive X direction. The cutter always starts in the positive Y direction on square pockets.



Fixed Cycles

Rectangular pocket milling: G75/G76



Process

The milling direction depends on the programming (here, G76). The maximum stepover is k .

The process is repeated until the programmed milling depth is reached.

On completion, the tool is withdrawn to the starting position.

Stepover

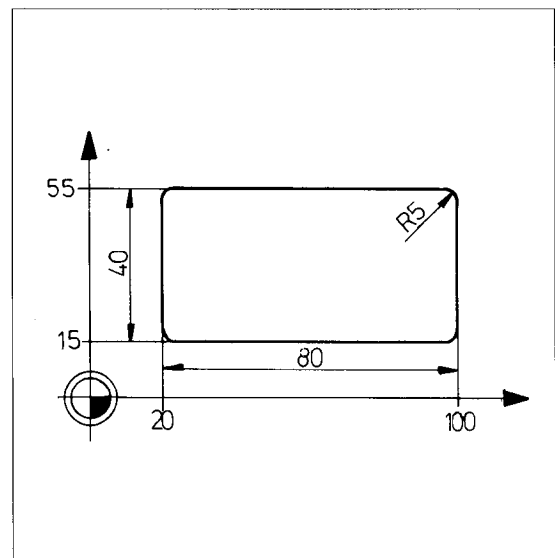
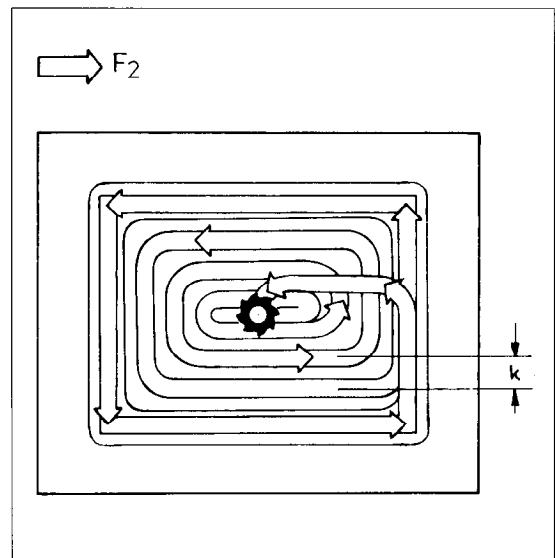
Stepover k is computed by the control according to the following formula:

$$k = F \times R$$

k : stepover

F : the overlap factor specified by the machine manufacturer (depends upon a machine parameter, see index A "General Information, MOD Functions, User parameters")

R : cutter radius



Example

G99 T1 L+0 R5 *
T1 G17 S200 *

G76 P01 -2
P02 -30
P03 -10
P04 80
P05 X+80
P06 X+40
P07 100 *

Setup clearance
Milling depth
Pecking depth
Feed rate for pecking

1st side length of the pocket
2nd side length of the pocket
Feed rate

G00 G40 X+45 Y+35 M3 *

Pre-positioning in X, Y,
spindle on

Z+2 M99 *

Pilot positioning in Z,
cycle call

Fixed Cycles

Circular pocket milling: G77/G78



The cycle

The circular pocket milling cycle is a roughing cycle.

Tool required

The cycle requires a center-cut end mill (ISO 1641) or pilot drilling at the pocket center S.

Input data

Setup clearance A: distance between tool tip (starting position) and workpiece surface.

Milling depth B (= pocket depth): distance between workpiece surface and bottom of pocket.

Pecking depth C: amount by which the tool penetrates the workpiece.

The signs for setup clearance, milling depth and pecking depth are all the same (usually negative).

Feed rate for pecking F_1 : traversing speed of the tool at penetration.

Circle radius R: radius of the circular pocket.

Feed rate F_2 : traversing speed of the tool in the machining plane.

Direction of the milling path:

Conventional milling (up cut)

G77: clockwise, up-cut milling with M3

Climb milling (down cut)

G78: counterclockwise, down-cut milling with M3

Starting position

The starting position S (pocket center) must be approached without radius compensation in a preceding positioning block.

Process

- The tool penetrates the work from the starting position (pocket center) at the "feed rate for pecking".
- The cutter then follows the programmed spiral path at feed rate F_2 . The direction of the path depends upon the programming (here, G78).

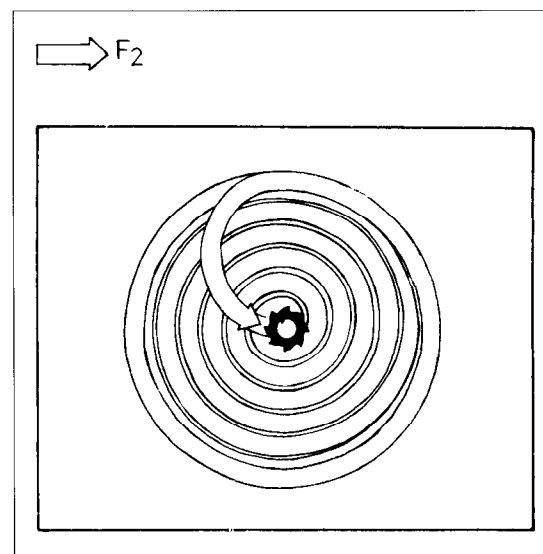
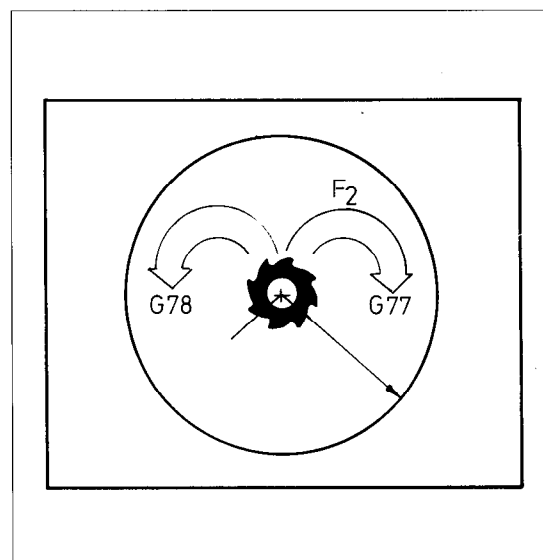
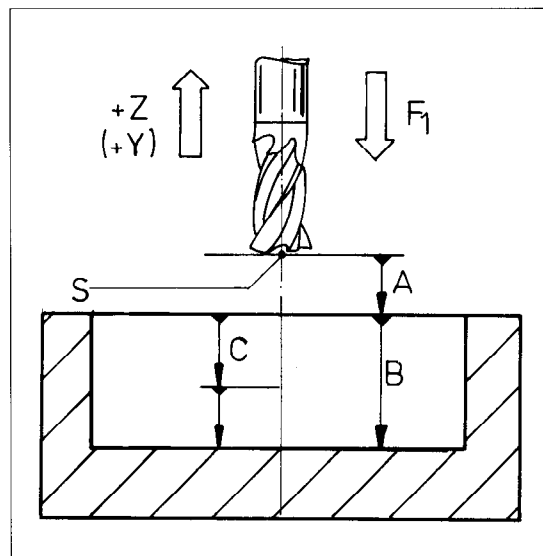
The starting direction of the cutter is for the

- XY plane the Y+ direction,
- ZX plane the X+ direction,
- YZ plane the Z+ direction.

The maximum stepover is the value k (see "Rectangular Pocket Milling").

The process is repeated until the programmed milling depth is reached.

When milling is completed, the tool is withdrawn to the starting position.



Fixed Cycles

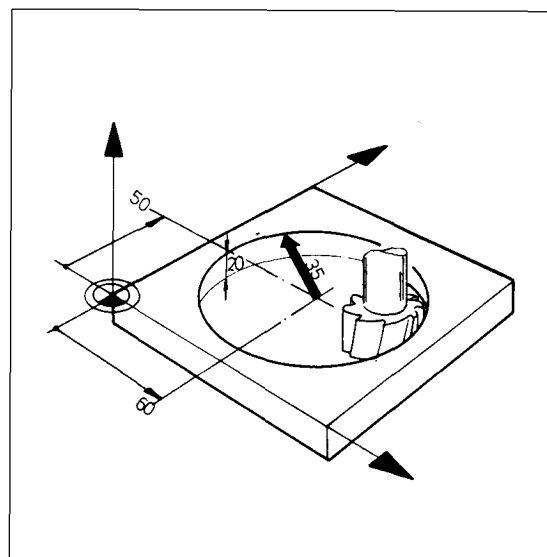
Circular pocket milling: G77/G78



Example

A circular pocket with radius 35 mm and depth 20 mm is to be milled at position X+60 Y+50.

G99 T1 L+0 R10 *
T1 G17 S200 *



G77 P01 -2
P02 -20
P03 -6
P04 80
P05 +35
P06 100 *

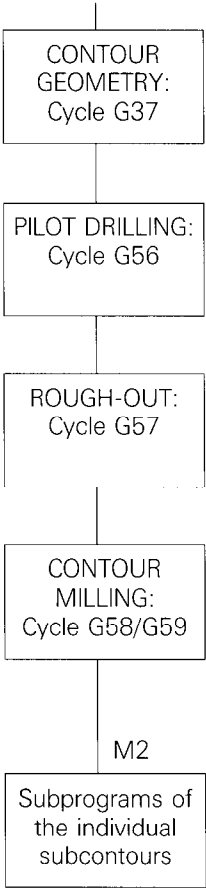
Setup clearance
Milling depth
Pecking depth
Feed rate for pecking
Circle radius
Milling feed rate

G00 G40 X+60 Y+50 M03 *

Pre-positioning in X and Y

Z+2 M99 *

Starting position in Z, cycle call



Scheme of a
program with
SL cycles

The group of cycles that we categorize as SL cycles is designed for efficient programming and milling of contours with one or more tools. The contour can be composed of several overlapping subcontours which are defined in separate subprograms.

The term SL cycles is derived from the characteristic **Subcontour List** of cycle G37 CONTOUR GEOMETRY, in which the list of subprograms is filed.

The control superimposes the separate contours to form a single whole. The programmer need not calculate the points of intersection!

To be able to work with several tools, the machining task is defined in cycle G37 without tool-specific data or feed values; those are entered in the individual cycles:

- G56** Pilot drilling (if required)
- G57** Rough-out
- G58/G59** Contour milling (finishing)

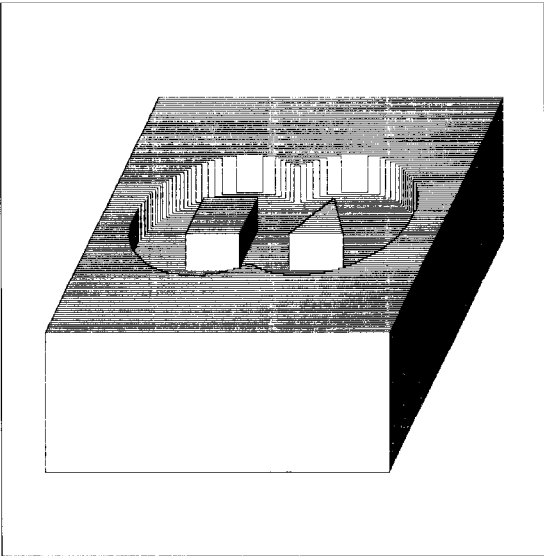
Each subprogram must specify whether **G41 or G42 radius compensation** applies and in which **direction** the contour is to be machined. The control deduces from these data whether the specific subprogram describes a **pocket** or an **island**.

- The control recognizes a pocket if the tool path lies inside the contour.
- The control recognizes an island if the tool path lies outside the contour.

- Be sure to run a graphic simulation before executing a program to see whether the contour was computed by the control as desired.
- All coordinate transformations are allowed in programming the contours (see "Coordinate Transformations, Overview").

Not all of the SL cycles are always required.

For easier familiarization, the following examples begin with only the rough-out cycle and then proceed progressively to the full range of functions.



SL Cycles

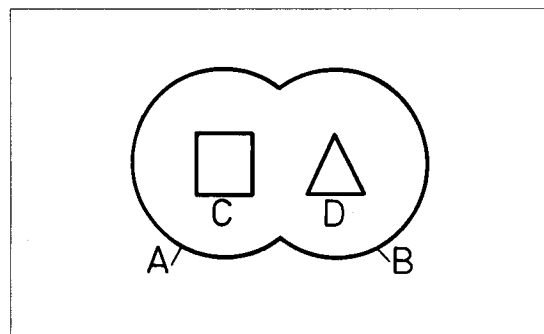
Contour geometry: G37

Rough-out: G57



Contour geometry: G37

The label numbers (subprograms) of the subcontours are specified in cycle G37 "contour geometry".
Up to 12 label numbers can be entered.
The TNC computes the intersections of the resulting contour from the subcontours.
Cycle G37 is immediately effective after definition (this cycle cannot be called).
The list of subcontours in cycle G37 should begin with a pocket.



A, B = Pockets
C, D = Islands

Example

N5 G37 P01 11 P02 12 P03 13 *

The subprograms 11, 12 and 13 define the complete contour in the example.

Rough-out: G57

Cycle G57 specifies the cutting path and partitioning.
It must be called, and can be executed separately.

Tool required

Cycle G57 requires a center-cut end mill (ISO 1641) if no pilot drilling is desired and if the tool must repeatedly jump over contours and plunge to the milling depth.

Input data

Setup clearance (A), milling depth (B), pecking depth (C) are incremental with the same signs (usually negative).

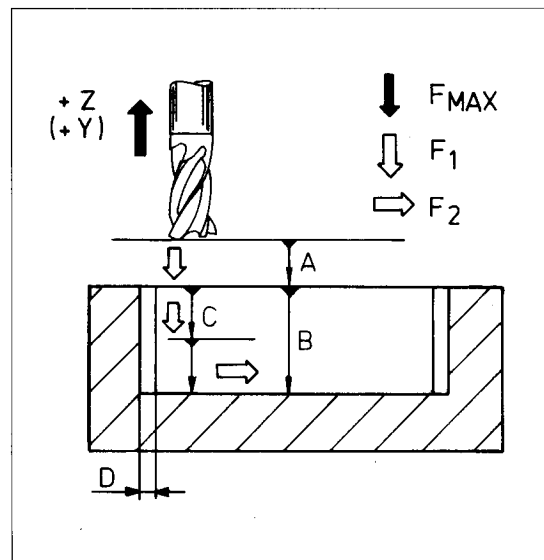
Feed rate for pecking: traversing speed of the tool at penetration (F1).

Finishing allowance: allowance in the machining plane, positive value (D).
If a negative allowance is entered, **pockets** will be milled **too large** by twice the allowance, while **islands** will be milled **too small** by the same amount.

Rough-out angle: roughing out direction relative to the reference axis of the machining plane.

Feed rate: traversing speed of the tool in the machining plane (F2).

The tool must be positioned at the setup clearance (A) before the cycle call.



Example

N16 G57 P01 -2
P02 -20
P03 -10
P04 40
P05 +1
P06 +0
P07 60 *

Setup clearance
Milling depth
Pecking depth
Feed rate for penetration
Finishing allowance
Rough-out angle
Feed rate in the working plane

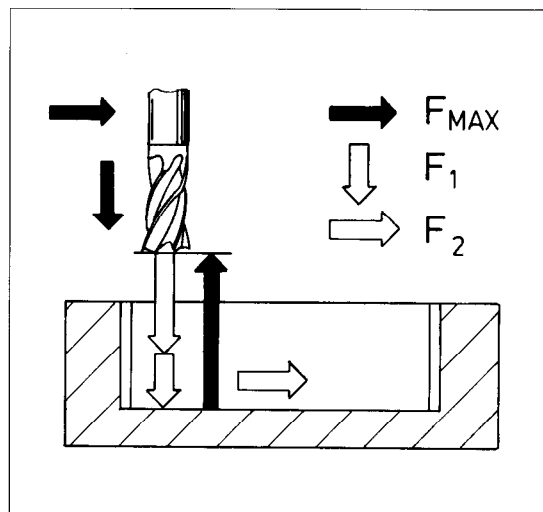
SL Cycles

Rough-out: G57



Process

The tool is automatically positioned over the first penetration point (with finishing allowance). It may be necessary to pre-position the tool before the call to prevent collision. The tool penetrates at the feed rate for pecking.



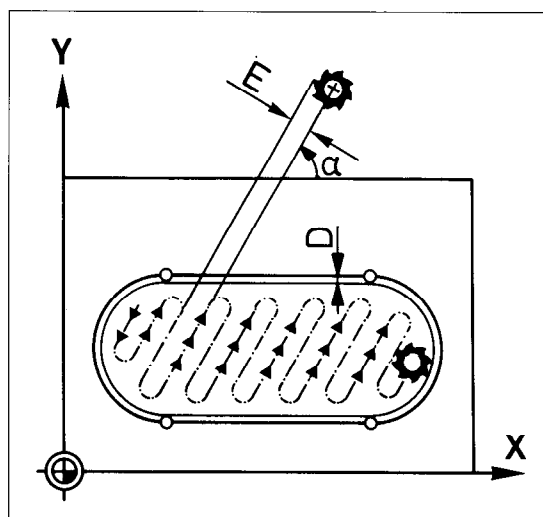
Milling the contour

After reaching the first pecking depth, the tool mills the first subcontour at the programmed milling feed rate with the finishing allowance.

At the penetration point, the tool is advanced to the next pecking depth. This process is repeated until the programmed milling depth is attained. Further subcontours are milled in the same manner.

Clearing the area

The area is then roughed out, the tool skipping over islands as follows: the tool retracts in rapid traverse to the setup clearance and moves to the next calculated penetration point. The tool then penetrates behind the island in the pre-milled channel at the feed rate for pecking. The feed direction corresponds to the programmed rough-out angle and can be set, so the resulting cuts are as long as possible with few cutting movements. The stepover equals the tool radius. Clearing out can be performed with multiple down-feeds.



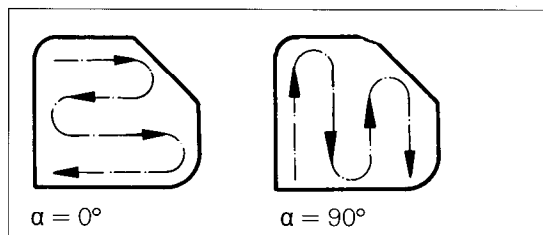
D = Finishing allowance
E = Stepover
 α = Rough-out angle

The tool is retracted to the setup clearance at the end of the cycle.

Sequence contour milling/area clearance

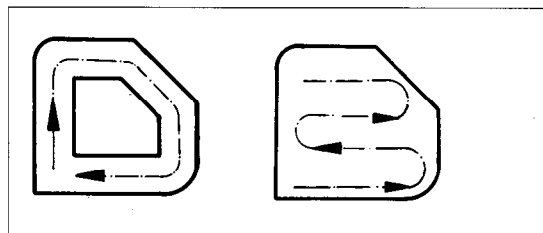
A machine parameter determines whether the contour is milled first and then the area cleared or vice versa.

In the same way is specified whether contour milling or roughing out is performed continuously over all infeeds, or for each infeed in the specified sequence.



Climb/conventional

A machine parameter also determines whether the contour is milled conventionally or by climb cutting (see index A "General Information, MOD Functions, User parameter MP 7420").



Begin with
contour
milling

Begin with
surface
clearing

SL Cycles

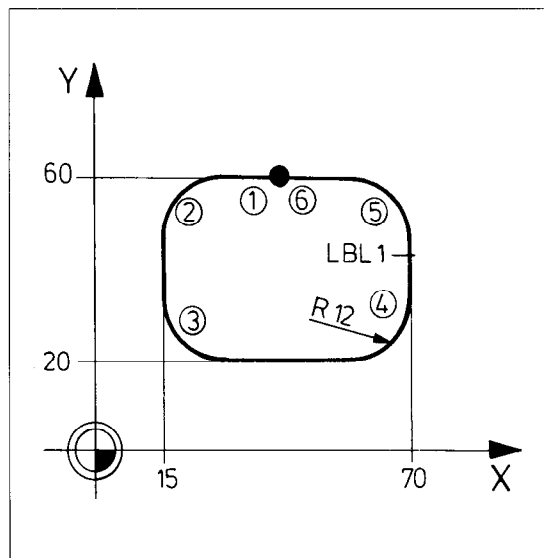
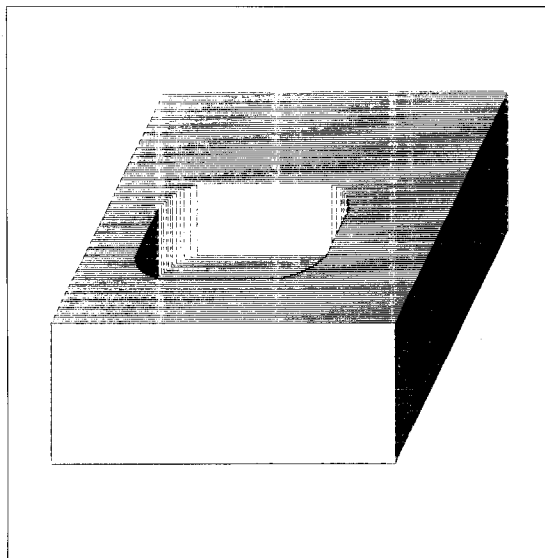
Roughing-out a rectangular pocket



Task

Rectangular pocket with rounding radius.

Interior machining of rectangular pocket with rounded corners, with a center-cut end mill (ISO 1641), tool radius 5 mm.



PGM %7206

```
%7206 G71 *
N10 G30 G17 X-20 Y-20 Z-40 *
N20 G31 G90 X+120 Y+120 Z+0 *
N30 G99 T1 L+0 R+5 *
N42 T1 G17 S1000 *
N50 G00 G90 Z+100 M03 *
```

Blank min. point
Blank max. point
Tool definition
Tool call

```
N60 G37 P02 1 *
N70 G57 P01 -2 P02 -20 P03 -8
      P04 100 P05 +0 P06 +0
      P07 500 *
```

"List" of contour subprograms
Definition for "rough-out"

```
N80 G40 X+40 Y+50 Z+2 M99 *
```

Pre-positioning, cycle call

```
N90 G00 G40 Z+20 M02 *
```

Retract, return jump to start of program

```
N100 G98 L1 *
N110 G41 X+40 Y+60 * ①
N120 X+15 * ②
N130 G25 R12 *
N140 Y+20 * ③
N150 G25 R12 *
N160 X+70 * ④
N170 G25 R12 *
N180 Y+60 * ⑤
N190 G25 R12 *
N200 X+40 * ⑥
N210 G98 L0 *
N9999 %7206 G71 *
```

Contour subprogram
Radius compensation is G41 (RL) and tool path is counterclockwise, the control therefore deduces:
pocket.

PGM %7207 creates a contour island with identical dimensions.

SL Cycles

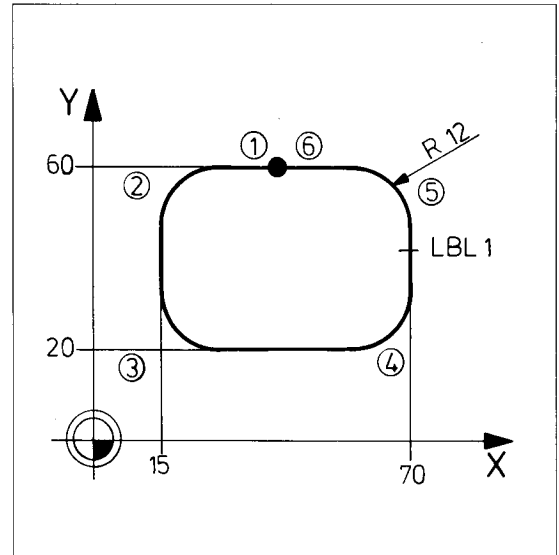
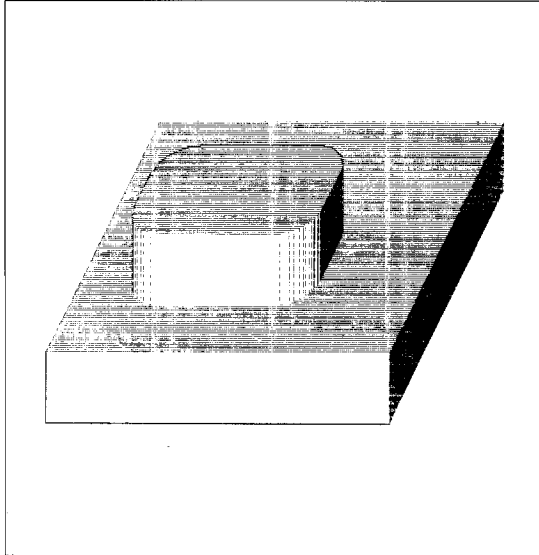
Roughing-out a rectangular island



Task

Rectangular island with rounding radius.

Exterior machining of rectangular island with rounded corners, with a center-cut end mill (ISO 1641), tool radius 5 mm.



PGM %7207

```
%7207 G71 *
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T1 L+0 R+5 *
N40 T1 G17 S111 *
N50 G00 G90 Z+100 M03 *

N60 G37 P01 2 P02 1 *
N70 G57 P01 -2 P02 -20 P03 -8
      P04 100 P05 +0 P06 +0
      P07 500 *
```

```
N80 G40 X+40 Y+50 Z+2 M99 *
```

```
N90 G00 G40 Z+20 M02 *
```

```
N100 G98 L1 *
N110 G42 X+40 Y+60 * ①
N120 X+15 * ②
N130 G25 R12 *
N140 Y+20 * ③
N150 G25 R12 *
N160 X+70 * ④
N170 G25 R12 *
N180 Y+60 * ⑤
N190 G25 R12 *
N200 X+40 * ⑥
N210 G98 L0 *
```

```
N220 G98 L2 *
N230 G41 X-5 Y-5 *
N240 X+105 *
N250 Y+105 *
N260 X-5 *
N270 Y-5 *
N280 G98 L0 *
N9999 %7207 G71 *
```

PGM %7206 creates a contour pocket with identical dimensions.

Blank

Tool

"List" of contour subprograms (sequence!)
Definition for "rough-out"

Pre-positioning, cycle call

Retract, return jump to start of program

Radius compensation is G42 (RR) and tool path is counterclockwise, the control therefore deduces: **island.**

Auxiliary pocket to externally limit the machined surface

SL Cycles Overlaps



**Overlapping
pockets and
islands**

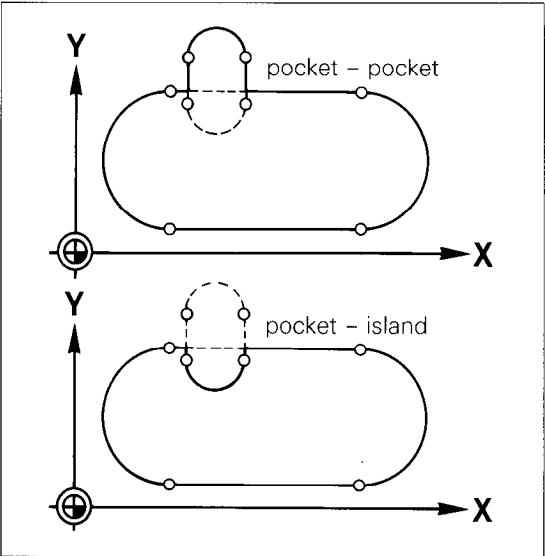
Pockets and islands can be overlapped (superimposed). The resulting contour is computed by the TNC.

The area of a pocket can, for example, be enlarged by an another pocket or reduced by an island.

**Starting
position**

Machining begins at the starting position of the first contour label of cycle G37. The starting positions should be located as far as possible from the superimposed contours.

If the subcontours are always defined in the same working direction, then for example with a positive working direction pockets can be easily recognized by the G41 (RL) compensation, and islands by the G42 (RR).



SL Cycles

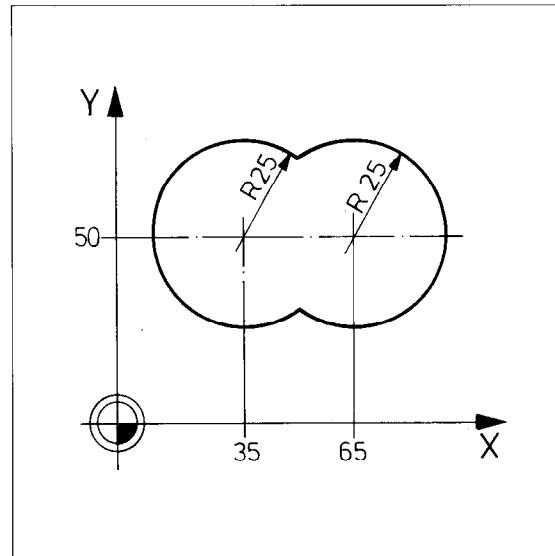
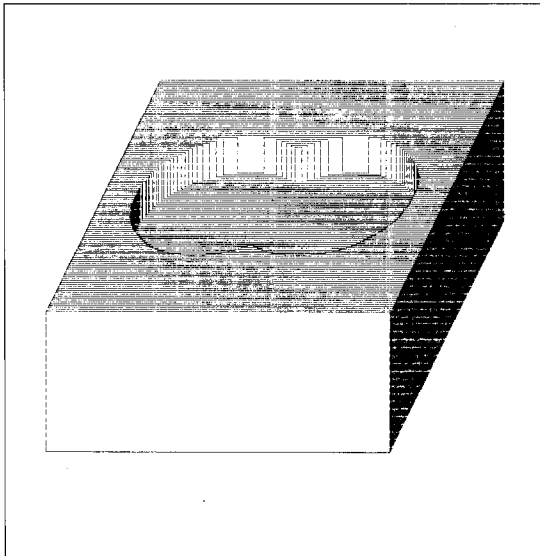
Overlapping pockets



Task

Overlapped pockets.

Interior machining of overlapping pockets with a center-cut end mill (ISO 1641), tool radius 3 mm.



PGM %7208

```
%7208 G71 *
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T2 L+0 R+3 *
N40 T2 G17 S100 *
N50 G00 G90 Z+200 *
N60 G40 X+50 Y+50 M03 *

N70 G37 P01 1 P02 2 *

N80 G57 P01 -2 P02 -10 P03 -10
    P04 500 P05 +0 P06 +0
    P07 500 *

N90 Z+2 M99 *

N100 G00 G40 Z+200 M02 *
```

Blank, tool axis

Tool

Pre-position X and Y, spindle on

"List" of contour subprograms

Definition for "rough-out"

Setup clearance Z, cycle call

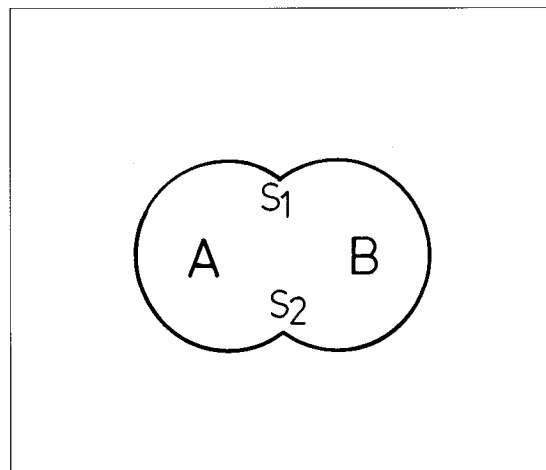
Retract return jump to start of program

Note

Machining begins with the first contour label defined in block N70!
The first pocket must begin outside the second pocket.

SL Cycles

Overlapping pockets



Points of intersection

The pocket elements A and B overlap each other. Since the control automatically computes the points of intersection S1 and S2, these points need not be programmed.

They are programmed as full circles.

```
N110 G98 L1 *
N120 G41 X+10 Y+50 *
N130 I+35 J+50 *
N140 G03 X+10 Y+50 *
N150 G98 L0 *
```

} A Left pocket

```
N160 G98 L2 *
N170 G01 G41 X+90 Y+50 *
N180 I+65 J+50 *
N190 G03 X+90 Y+50 *
N200 G98 L0 *
```

} B Right pocket

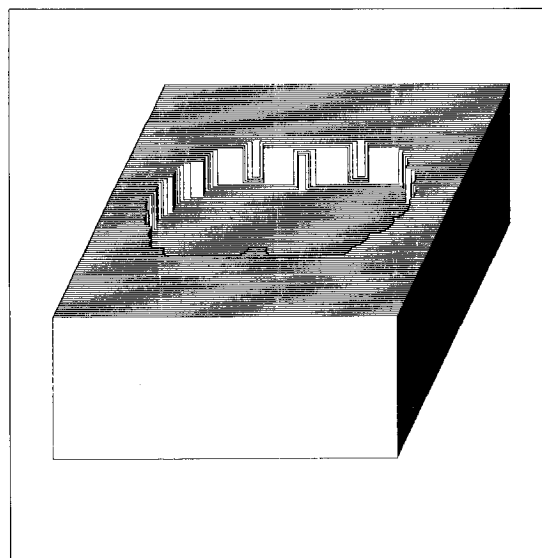
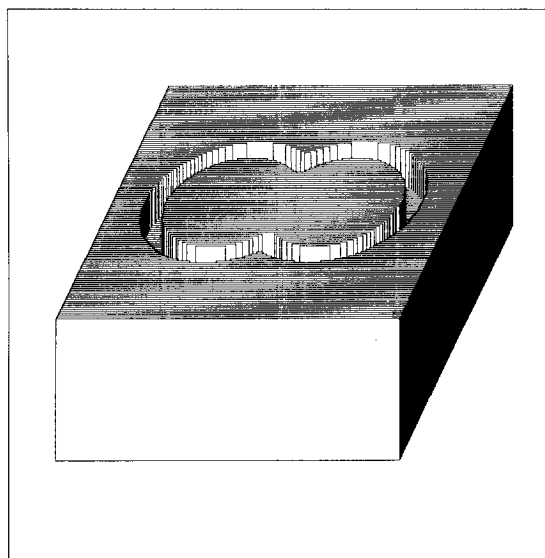
```
N9999 %7208 G71 *
```

Execution

Depending on the control setup (machine parameters), machining begins either with the contour edge or the area.

Contour edge is machined first

Area is machined first



SL Cycles

Overlapping pockets



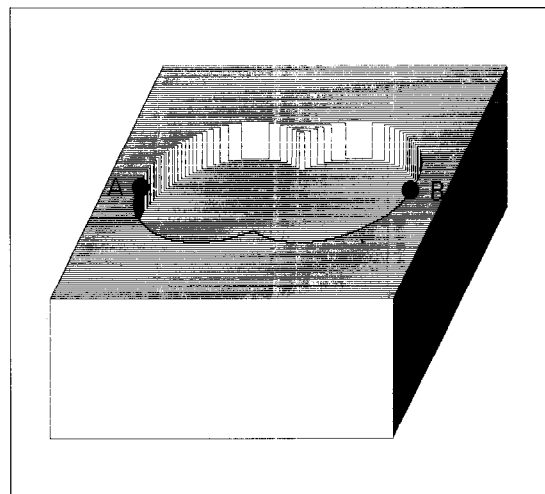
"Sum" area

Both areas (element A and element B) along with the common overlapping area are to be machined.

- A and B must be pockets.
- the first pocket (in cycle G37) must begin outside the second.

```
N110 G98 L1 *
N120 G41 X+10 Y+50 *
N130 I+35 J+50 *
N140 G03 X+10 Y+50 *
N150 G98 L0 *
```

```
N160 G98 L2 *
N170 G01 G41 X+90 Y+50 *
N180 I+65 J+50 *
N190 G03 X+90 Y+50 *
N200 G98 L0 *
```



- A and B are the starting points of the contour labels.

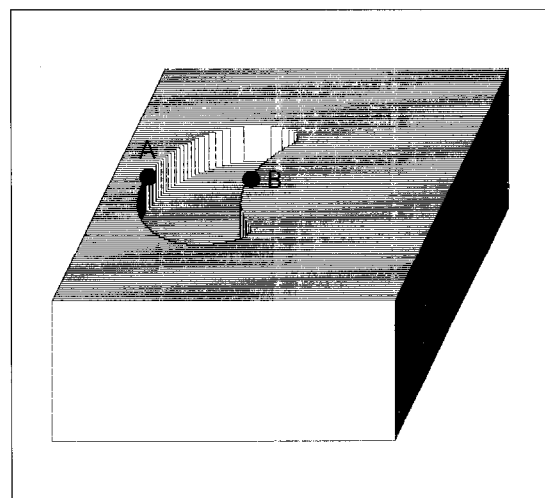
"Difference" area

Area A is to be machined without the portion overlapped by B:

- A must be a pocket and B an island.
- A must begin outside of B.

```
N110 G98 L1 *
N120 G41 X+10 Y+50 *
N130 I+35 J+50 *
N140 G03 X+10 Y+50 *
N150 G98 L0 *
```

```
N160 G98 L2 *
N170 G01 G42 X+90 Y+50 *
N180 I+65 J+50 *
N190 G03 X+90 Y+50 *
N200 G98 L0 *
```



An island can also reduce several pocket areas. The starting points of the pocket contours must all be outside the island.

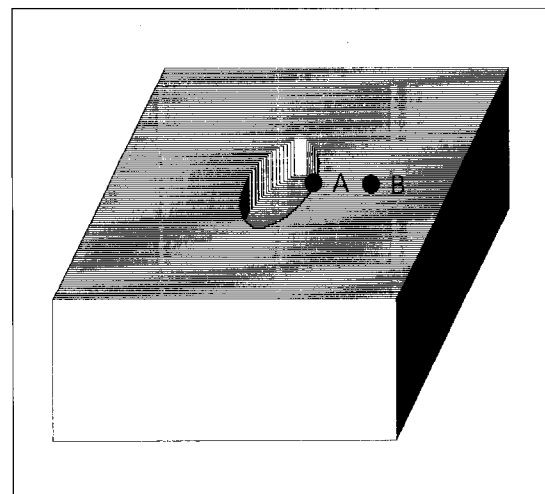
"Intersecting" area

Only the area covered commonly by A and B is to be machined.

- A and B must be pockets.
- A must begin inside of B.

```
N110 G98 L1 *
N120 G41 X+60 Y+50 *
N130 I+35 J+50 *
N140 G03 X+60 Y+50 *
N150 G98 L0 *
```

```
N160 G98 L2 *
N170 G01 G41 X+90 Y+50 *
N180 I+65 J+50 *
N190 G03 X+90 Y+50 *
N200 G98 L0 *
```



SL Cycles

Overlapping islands



Expanding program %7208

N70 G37 P01 1 P02 2 P03 5 *

N210 G98 L5 *
N220 G01 G41 X+5 Y+5 *
N230 X+95 *
N240 Y+95 *
N250 X+5 *
N260 Y+5 *
N270 G98 L0 *

An island always requires an additional outer limit = pocket (here, G98 L5).

A pocket can also reduce several island areas. This pocket must begin inside the first island. The starting points of the remaining intersected island contours must be outside the pocket.

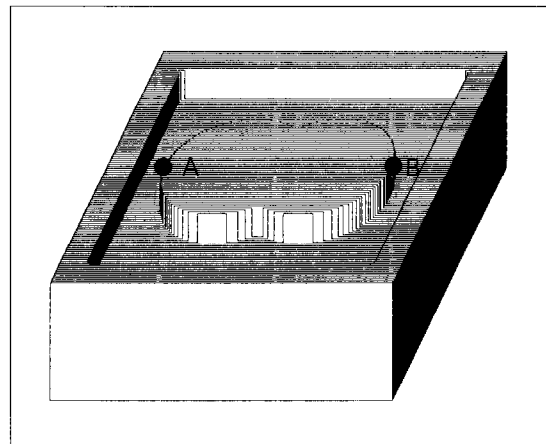
"Sum" area

Both areas (element A and element B) along with the common overlapping area are to remain unmachined.

- A and B must be islands.
- The first island must begin outside the second.

N110 G98 L1 *
N120 G42 X+10 Y+50 *
N130 I+35 J+50 *
N140 G03 X+10 Y+50 *
N150 G98 L0 *

N160 G98 L2 *
N170 G01 G42 X+90 Y+50 *
N180 I+65 J+50 *
N190 G03 X+90 Y+50 *
N200 G98 L0 *



• A, • B are the starting points of the subcontours.

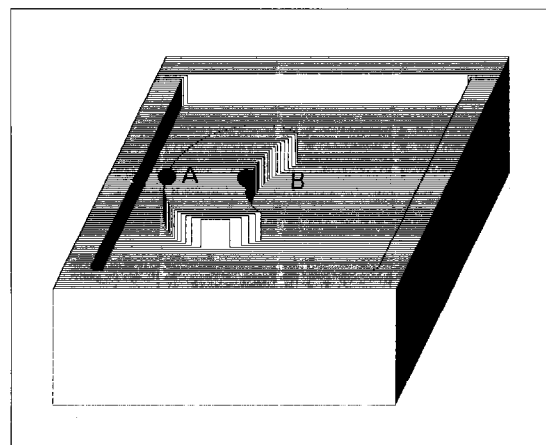
"Difference" area

Area A is to remain unmachined except that portion overlapped by B.

- A must be an island and B a pocket.
- A must begin outside of B.

N110 G98 L1 *
N120 G42 X+10 Y+50 *
N130 I+35 J+50 *
N140 G03 X+10 Y+50 *
N150 G98 L0 *

N160 G98 L2 *
N170 G01 G41 X+40 Y+50 *
N180 I+65 J+50 *
N190 G03 X+40 Y+50 *
N200 G98 L0 *



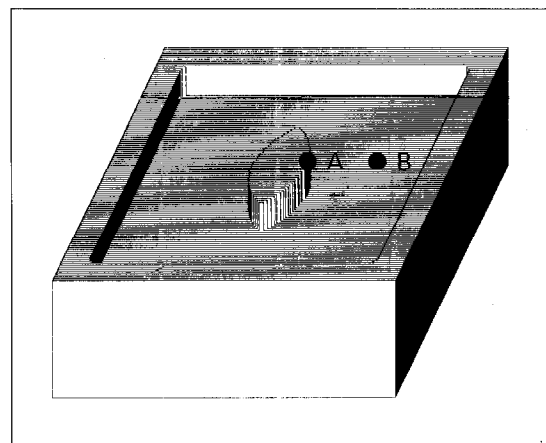
"Intersecting" area

Only the area covered commonly by A and B remains unmachined.

- A and B must be islands.
- A must begin inside of B.

N110 G98 L1 *
N120 G42 X+60 Y+50 *
N130 I+35 J+50 *
N140 G03 X+60 Y+50 *
N150 G98 L0 *

N160 G98 L2 *
N170 G01 G42 X+90 Y+50 *
N180 I+65 J+50 *
N190 G03 X+90 Y+50 *
N200 G98 L0 *



SL Cycles

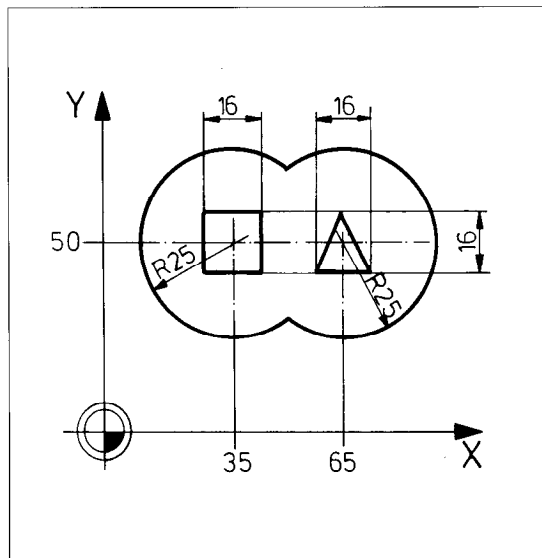
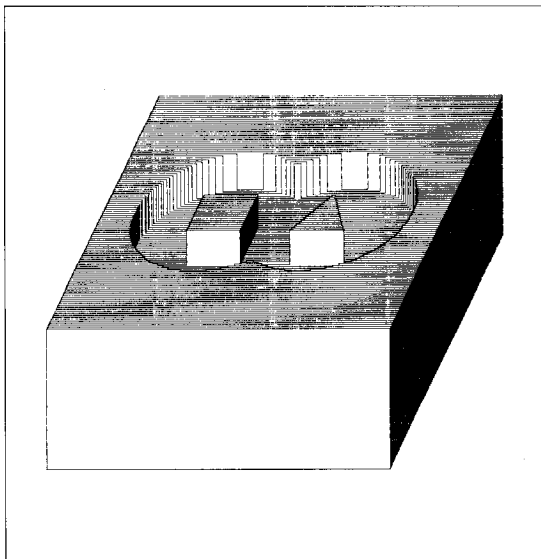
Overlapping pockets and islands



Task

Overlapping pockets with islands.
Island within a pocket.

Interior machining of overlapping pockets and islands with a center-cut end mill (ISO 1641), tool radius 3 mm. Islands are located within a pocket area.



Main program %7209

```
%7209 G71 *
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
N30 G99 T2 L+0 R+2.5 *
N40 G37 P01 1 P02 2 P03 3 P04 4 *
N50 G98 L10 *
N60 T0 G17 *
N70 G00 G40 G90 Z+20 *
N80 X-20 Y-20 *
N90 G98 L0 *
N100 M06 *
N110 T2 S100 *

N120 G57 P01 -2 P02 -10 P03 -5
      P04 500 P05 +0 P06 +0 P07 500 *

N130 Z+2 *
N140 G79 M03 *
N150 L10,0 *
N160 G00 Z+20 M02 *
```

List of contour elements

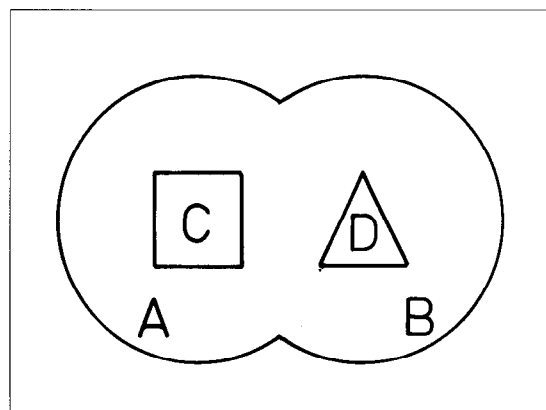
Program %7209 is an expansion of program %7208: the interior islands are added (subprograms 3 and 4).

SL Cycles

Overlapping pockets and islands



The entire contour is composed of the elements A and B, i.e. two overlapping pockets and C and D, i.e. two islands within these pockets.



**Contour
subprograms
for program
%7209**

N170 G98 L1 *
N180 G41 X+35 Y+25 *
N190 I+35 J+50 *
N200 G03 X+35 Y+25 *
N210 G98 L0 *

N220 G98 L2 *
N230 G01 G41 X+65 Y+25 *
N240 I+65 J+50 *
N250 G03 X+65 Y+25 *
N260 G98 L0 *

N270 G98 L3 *
N280 G01 G42 X+35 Y+42 *
N290 X+43 *
N300 Y+58 *
N310 X+27 *
N320 Y+42 *
N330 X+35 *
N340 G98 L0 *

N350 G98 L4 *
N360 G01 G42 X+65 Y+42 *
N370 X+73 *
N380 X+65 Y+58 *
N390 X+57 Y+42 *
N400 X+65 *
N410 G98 L0 *
N9999 %7209 G71 *

} A Left pocket

} B Right pocket

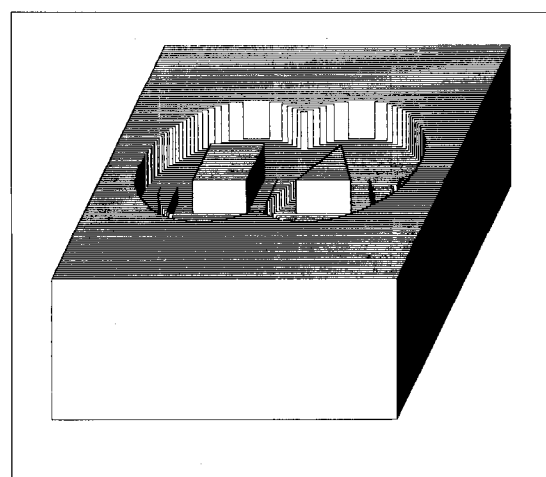
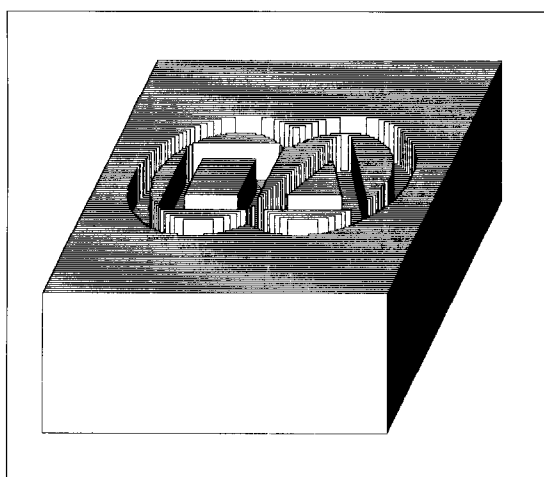
} C Square island

} D Triangular island

Execution

Machining of the contour edges

Area clearance (unfinished)



SL Cycles

Pilot drilling: G56

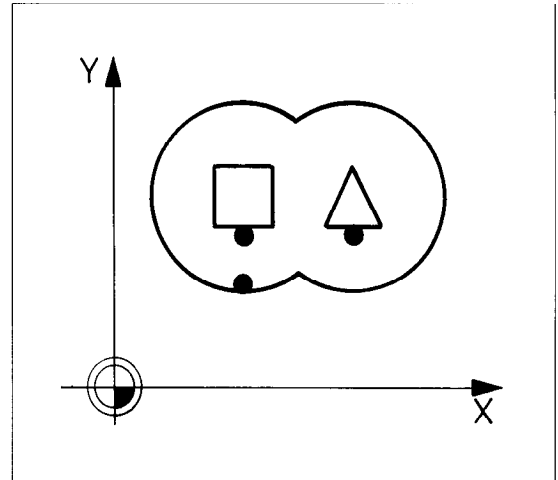


The cycle

Pilot drill the cutter infeed points at the starting points of the subcontours, compensated by the finishing allowance.

For closed contour sequences resulting from multiple superimposed pockets and islands, the infeed point is the starting point of the first sub-contour.

This cycle must be called!



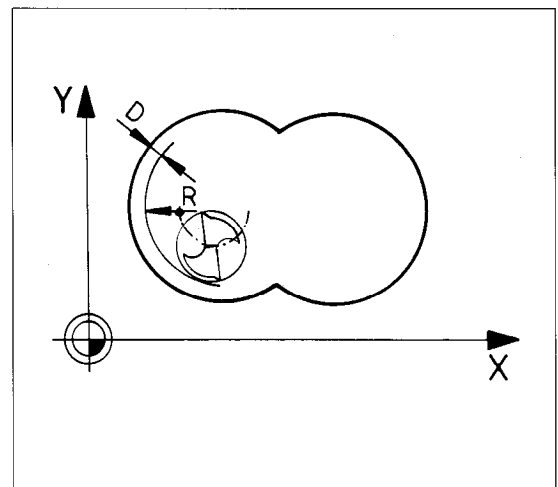
● Cutter infeed point

Input data

The input values are identical to pecking; enter a **finishing allowance** in addition.

Finishing allowance: allowance for drilling (positive value), effective in the working plane. The sum of the tool radius and the finishing allowance should be the same for pilot drilling and roughing-out.

The tool must be at the setup clearance before calling the cycle!



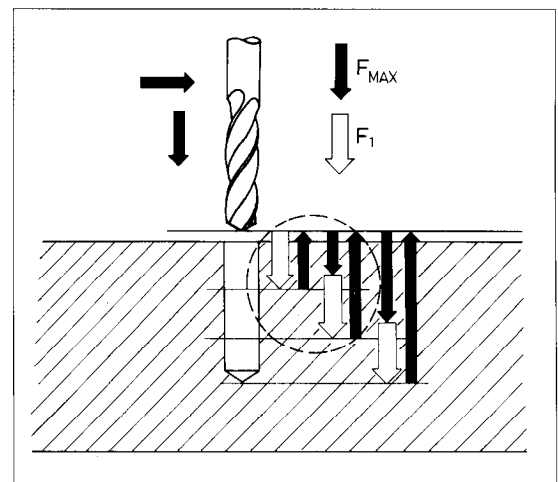
D = Finishing allowance
R = Tool radius

Process

The tool is automatically positioned over the first infeed point, offset by the allowance. The tool may have to be pre-positioned to prevent collision!

The drilling process is identical to the fixed cycle "pecking" (cycle 1).

Subsequently, the tool is positioned over the second infeed point at the programmed setup clearance, and the drilling procedure is repeated.



Example

N25 G56 P01 -2
P02 -20
P03 -10
P04 40
P05 +1 *

Setup clearance
Drilling depth
Pecking depth
Feed rate for infeed
Finishing allowance



The cycle

Cycle G58/G59 "contour milling" is used for finishing the contour pocket.

The cycle can also be generally used to mill contours made up of subcontours.

This offers the following benefits:

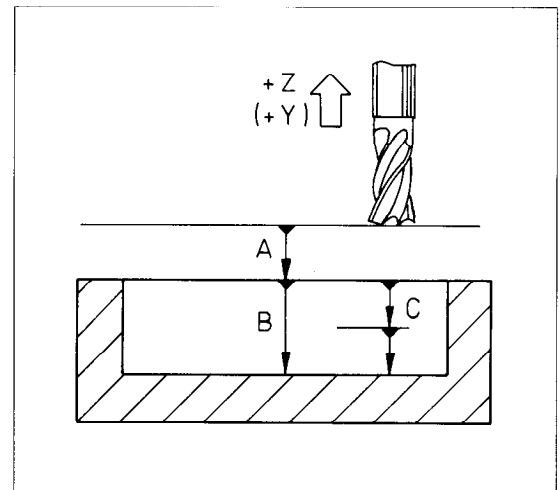
- contour intersections are computed,
- collisions are avoided.

Tool required

The cycle requires a center cutting tool.

The cycle must be called!

The setup clearance A, milling depth B and pecking depth C are identical to pecking. The signs must be the same (normally negative).



Input data

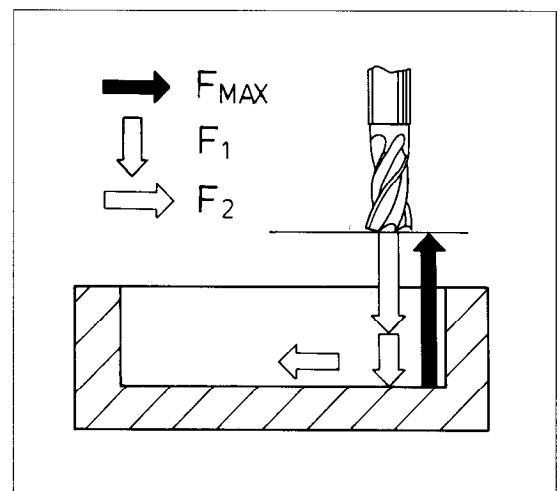
Feed rate for pecking: tool traversing speed at infeed (F_1).

Rotating direction for contour milling: milling direction along the pocket contour (island contours: opposite milling direction).

For the following directions, M3 means G58: down-cut milling for pocket and island, G59: up-cut milling for pocket and island.

Feed rate F_2 : tool traversing speed in the machining plane.

The tool must be at the setup clearance (A) prior to the cycle call.



Process

The tool is automatically positioned over the first contour point.

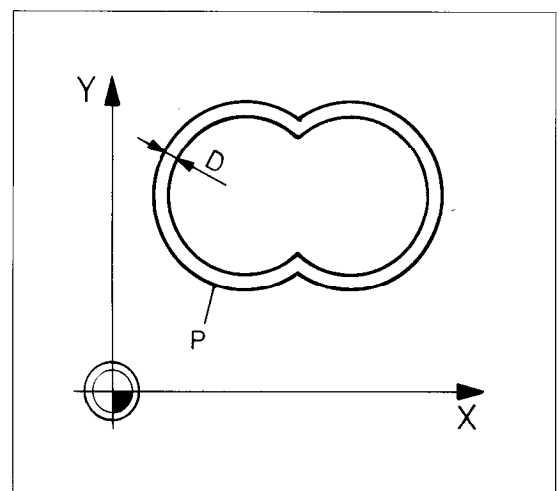
Beware of collisions with clamping devices!

The tool then penetrates the workpiece at the programmed **feed rate** to the first **pecking depth**.

After reaching the first pecking depth, the tool mills the first contour at the programmed **feed rate** in the specified **rotating direction**.

At the infeed point, the tool is advanced to the next pecking depth. The procedure is repeated until the programmed **milling depth** is attained.

The next subcontours are milled in the same manner.



P = Programmed contour (pocket)
D = Finishing allowance from cycle G57 rough-out

Example

N25 G58 P01 -2
P02 -20
P03 -10
P04 +40
P05 60 *

Setup clearance
Milling depth
Pecking depth
Feed rate for infeed
Feed rate in the working plane

SL Cycles

Machining with several tools



The following scheme illustrates the application of the SL cycles pilot drilling, rough-out, and contour milling in one program:

List of contour subprograms

Cycle definition:
with G37
No call!

Drilling

Define and call the drill
Cycle definition:
with G56
Pre-positioning,
Cycle call!

Rough-out

Define and call the roughing cutter
Cycle definition:
with G57
Pre-positioning,
Cycle call!

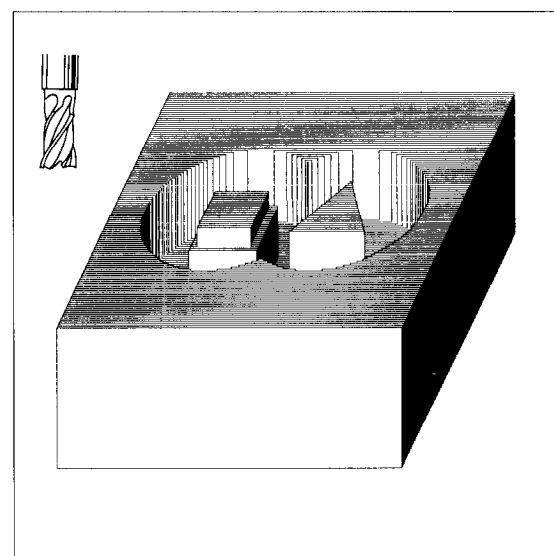
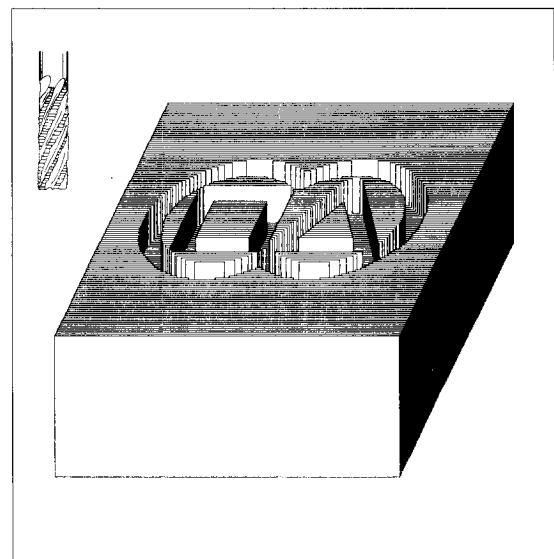
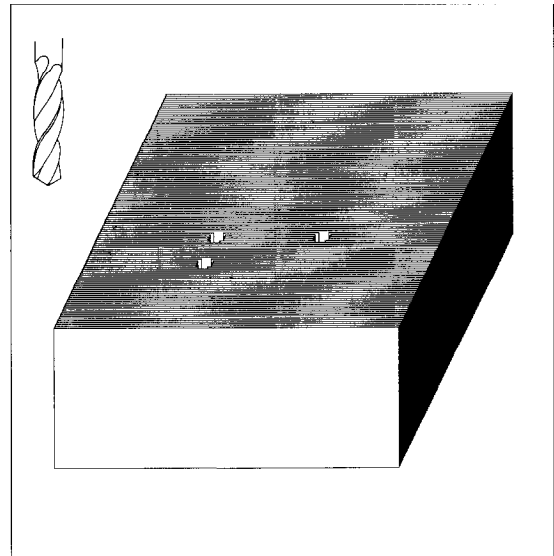
Finishing

Define and call the finishing cutter
Cycle definition:
with G58 or G59
Pilot positioning,
Cycle call!

Contour subprograms

M02

Subprograms for the subcontours



SL Cycles

Machining with several tools



Task	Overlapping pockets with islands.	
	Interior machining with pilot drilling, roughing, finishing.	
Main program	%7210 G71 *	
%7210	N10 G30 G17 X+0 Y+0 Z-40 *	
	N20 G31 G90 X+100 Y+100 Z+0 *	
	N30 G99 T1 L+0 R+2.2 *	Drill
	N40 G99 T2 L+0 R+3 *	Roughing cutter
	N50 G99 T3 L+0 R+2.5 *	Finishing cutter
	N60 G37 P01 1 P02 2 P03 3 P04 4 *	
	N70 G98 L10 *	
	N80 T0 G17 *	Tool change
	N90 G00 G90 Z+20 *	
	N100 G40 X-20 Y-20 *	
	N110 G98 L0 *	
	N120 M06 *	
	N130 T1 G17 S100 *	
	N140 G56 P01 -2 P02 -20 P03 -5 P04 500 P05 +2 *	Pilot drilling
	N150 Z+2 *	
	N160 G79 M03 *	
	N170 L10,0 *	
	N180 M06 *	
	N190 T2 G17 S100 *	
	N200 G57 P01 -2 P02 -20 P03 -5 P04 500 P05 +2 P06 +0 P07 500 *	Roughing-out
	N210 Z+2 *	
	N220 G79 M03 *	
	N230 L10,0 *	
	N240 M06 *	
	N250 T3 G17 S500 *	
	N260 G59 P01 -2 P02 -20 P03 -5 P04 100 P05 500 *	Finishing
	N270 Z+2 *	
	N280 G79 M03 *	
	N290 L10,0 *	
	N300 G00 G40 Z+20 M02 *	Retract and return jump to beginning of program.
Subprogram	N305 G98 L1 *	Left pocket
	N310 G41 X+35 Y+25 *	
	N320 I+35 J+50 *	
	N340 G03 X+35 Y+25 *	
	N350 G98 L0 *	
	N360 G98 L2 *	Right pocket
	N370 G01 G41 X+65 Y+25 *	
	N380 I+65 J+50 *	
	N390 G03 X+65 Y+25 *	
	N400 G98 L0 *	
	N410 G98 L3 *	Square island
	N420 G01 G42 X+35 Y+42 *	
	N430 X+43 *	
	N440 Y+58 *	
	N450 X+27 *	
	N460 Y+42 *	
	N470 X+35 *	
	N480 G98 L0 *	
	N490 G98 L4 *	Triangular island
	N500 G01 G42 X+65 Y+42 *	
	N510 X+73 *	
	N520 X+65 Y+58 *	
	N530 X+57 Y+42 *	
	N540 X+65 *	
	N550 G98 L0 *	
	N9999 %7210 G71 *	

The contour subprograms 1 to 4 are identical to those in program %7209.

Coordinate Transformations

Overview



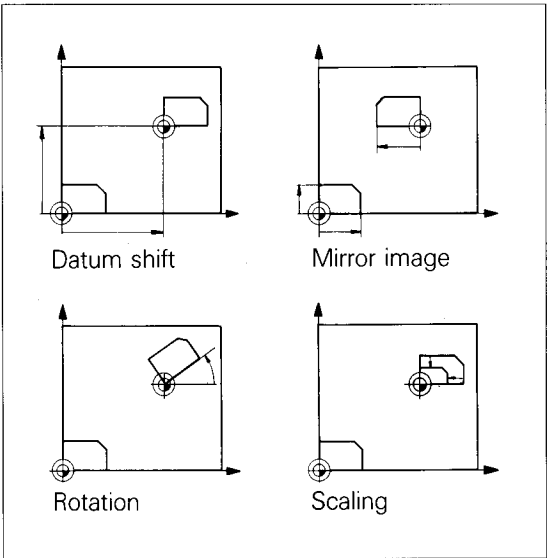
The following cycles serve for coordinate transformations:

G54 Datum shift
G28 Mirror image
G73 Rotation
G72 Scaling

Original

With the help of coordinate transformations, a program section can be executed as a variant of the "original".

In the following descriptions, subprogram 1 is always the "original" subprogram (identified by the gray background).



Immediate activation

Every transformation is immediately active – without being called.

Duration of activation

A coordinate transformation remains active until it is changed or cancelled.

Its effect is not impaired by interrupting and aborting program run. This is also true when the same program is restarted from another location with "GOTO □".

End of activation

You can cancel coordinate transformations in the following ways:

- Cycle definition for original condition (e.g.: scaling factor 1.0);
- Selecting another program with "PGM NR" in the operating mode program run "full sequence" or "single block".
- Programming of miscellaneous functions M02 or M30, or block N9999% ... (depending on the machine parameters);

Error message

CYCL INCOMPLETE

This error message is displayed if a fixed cycle is called after defining a transformation but no fixed cycle was defined. Otherwise the control executes the fixed cycle which was last defined.

Coordinate Transformations

Datum shift: G54



The cycle

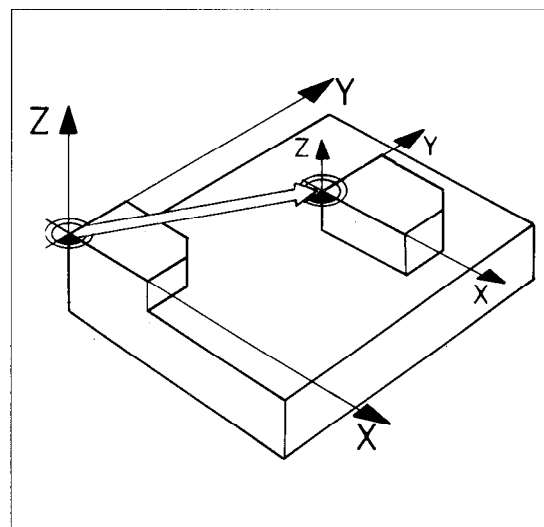
You can program a datum shift (also known as zero offset) to any point within a program. The manually set absolute workpiece datum remains unchanged.

Thus, identical machining steps (e.g. subprograms) can be executed at different positions on the workpiece without having to reenter the program section each time.

Combining with other coordinate transformations

If a datum shift is to be combined with other transformations, the shift usually has to be made **before** the other transformations.

In this way you can execute a program section at several locations and in modified form, such as rotated, reduced or mirrored.



Effect

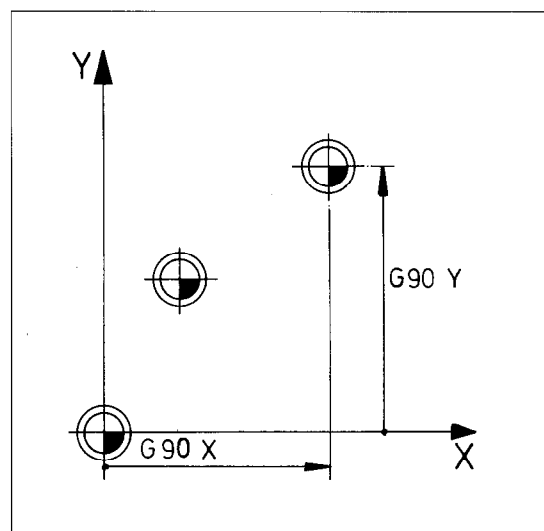
For a datum shift definition, only the coordinates of the new datum are to be entered.

An active datum shift is displayed in the status field. All coordinate inputs then refer to the new datum.

Incremental/absolute

In the cycle definition the coordinates can be entered as absolute or incremental dimensions:

- **Absolute:** The coordinates of the new datum refer to the manually set workpiece datum. Refer to the center figure.
- **Incremental:** The coordinates of the new datum refer to the last valid datum, which can itself be shifted. Refer to the lower figure.



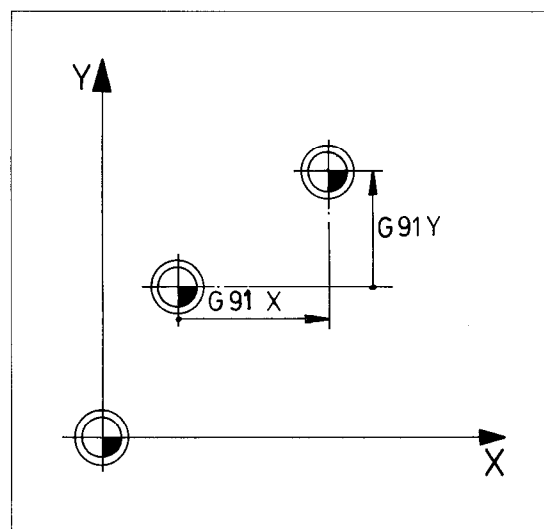
Absolute datum shift

Cancelling the shift

A datum shift is cancelled by entering the datum shift X0/Y0/Z0 ...

Only the "shifted" axes have to be entered.

G54 X+0 Y+0 Z+0 *



Incremental datum shift

Coordinate Transformations

Datum shift: G54



Selecting the cycle

Input

G 5 4



X

☐

Select the axis and coordinate of the new datum.

Y

☐

The datum shift is possible in all four axes.

...

END

Conclude block.

Example

A machining task is to be carried out as a subprogram

a) referenced to the set datum X+0/Y+0 and

b) additionally referenced to the shifted datum X+40/Y+60.

%54 G71 *

N10 G30 G17 X+0 Y+0 Z-40 *

N20 G31 G90 X+100 Y+100 Z+0 *

N30 G99 T1 L+0 R+5 *

N40 T1 G17 S200 *

N50 L1,0 *

Without datum shift ①

N60 G54 X+40 Y+60 *

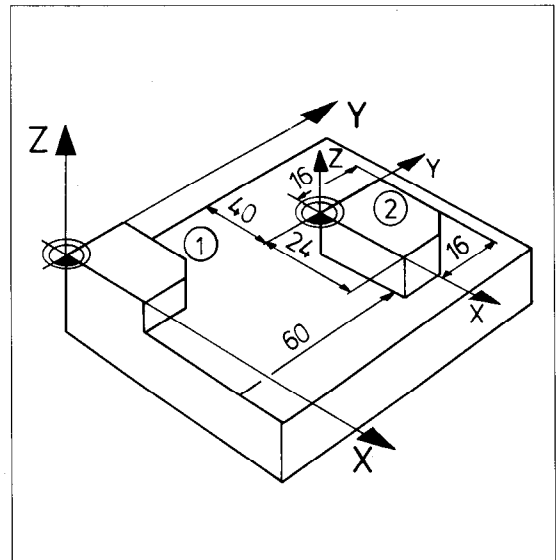
N70 L1,0 *

With datum shift ②

N80 G54 X+0 Y+0 *

Datum shift reset

N90 G00 Z+50 M02 *



Subprogram

N100 G98 L1 *

N110 G00 G40 X-10 Y-10 M03 *

N120 Z+2 *

N130 G01 Z-5 F100 *

N140 G41 X+0 Y+0 F500 *

N150 Y+20 *

N160 X+25 *

N170 X+30 Y+15 *

N180 Y+0 *

N190 X+0 *

N200 G40 X-10 Y-10 *

N210 G00 Z+2 *

N220 G98 L0 *

N9999 %54 G71 *

Coordinate Transformations

Mirror image: G28



The cycle

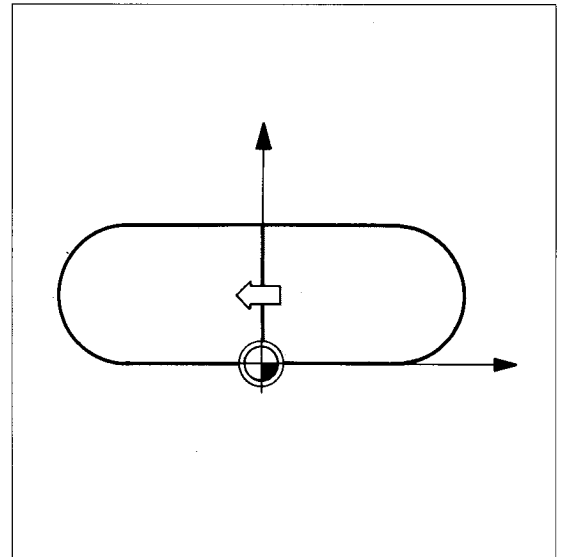
The direction of an axis is reversed when it is mirrored. The sign is reversed for all coordinates of this axis. The result is a mirror image of a programmed contour or of a hole pattern.

Mirroring is only possible in the working plane. You can mirror in one axis or both axes simultaneously.

Activation

The mirror image is immediately active upon definition. The mirrored axes can be recognized by the highlighted axis designations in the status display for the datum shift.

Mirroring is performed at the current datum. The datum must therefore be shifted to the required position before a "mirror image" cycle definition.



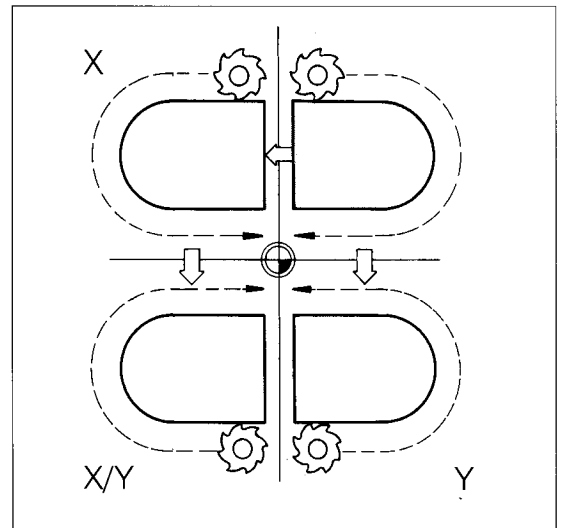
Mirrored axes

Enter the axis or axes to be mirrored. The tool axis cannot be mirrored.

Climb and conventional milling

Mirroring one axis: The rotating direction is changed with the coordinate signs, so climb milling becomes conventional and vice versa. The milling direction remains unchanged for fixed cycles.

Mirroring two axes: The contour which was mirrored in one axis is mirrored a second time – in the other axis. The direction of rotation and milling (climb or conventional) remains the same.



X, Y = Axes to be mirrored

Datum position

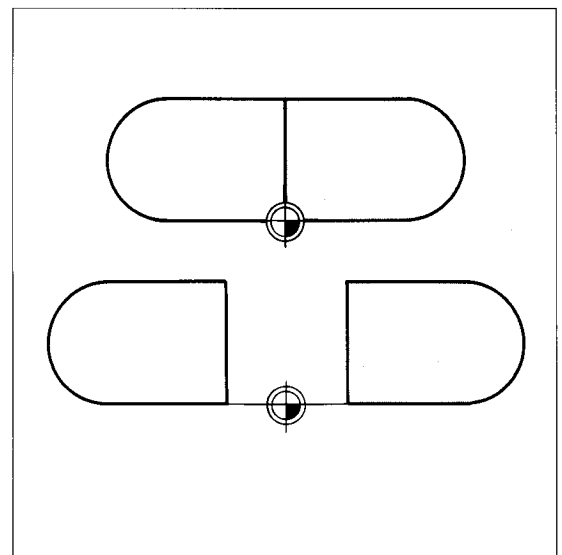
The position of the datum is very important for obtaining the desired change.

1. If the datum is on the part contour, the part "flips" over its own axis.
2. If the datum is outside the contour, the part "flips and jumps" to another position!

Cancelling the mirror image

The mirror image cycle is cancelled by entering the mirror image cycle and responding to the dialog query with "END ☐":

G28 *



Coordinate Transformations

Mirror image: G28



Selecting the cycle

Initiate the dialog



MIRROR IMAGE AXIS ?



Enter the axis to be mirrored, e.g. X.



Enter the second axis to be mirrored if applicable, e.g. Y.



Conclude block.

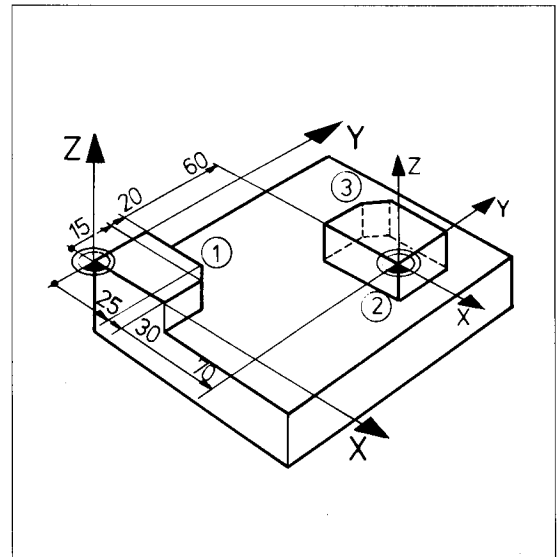
Example

A program section (subprogram 1) is to be executed – as originally programmed – at position X+0/Y+0. It is then mirrored in X and executed at the position X+70/Y+60.

```
%34 G71 *
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *

N30 G99 T1 L+0 R+5 *
N40 T1 G17 S200 *

N50 L1,0 *           Not mirrored ①
N60 G54 X+70 Y+60 *   Datum shift ②
N65 G28 X *          Mirror image ③
N70 L1,0 *           Subprogram call
N80 G54 X+0 Y+0 *     Cancel datum shift
N85 G28 *             Reset mirror image
N90 G00 G40 Z+50 M02 * Retract, return jump
```



Subprogram:

```
N100 G98 L1 *
N110 G00 G40 X-10 Y-10 M03 *
N120 Z+2 *
N130 G01 Z-5 F100 *
N140 G41 X+0 Y+0 F500 *
N150 Y+20 *
N160 X+25 *
N170 X+30 Y+15 *
N180 Y+0 *
N190 X+0 *
N200 G40 X-10 Y-10 *
N210 G00 Z+2 *
N220 G98 L0 *

N9999 %34 G71 *
```

Note

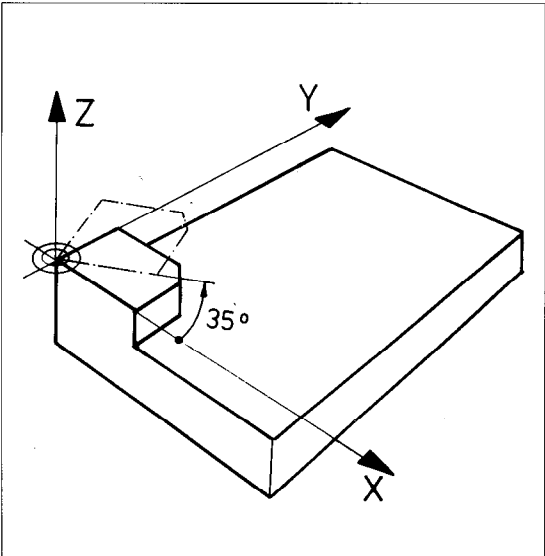
For correct machining according to the drawing, it is absolutely necessary that the sequence of cycles shown in the above execution be retained!

Coordinate Transformations

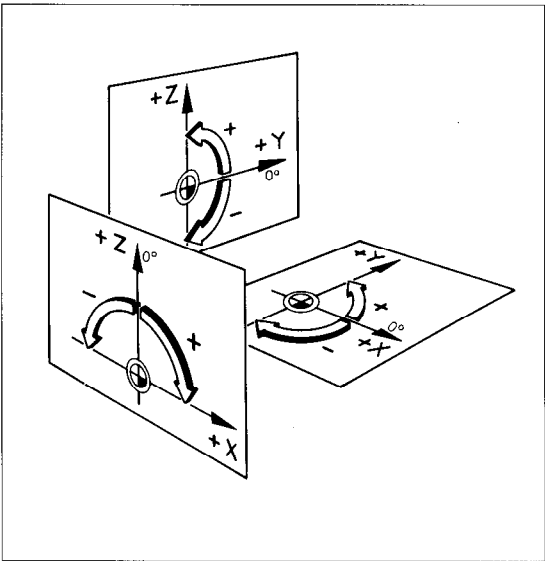
Coordinate system rotation: G73



- The cycle** The coordinate system can be rotated in the machining plane around the current datum in a program.
- Activation** Rotation is effective without being called and is also active in the operating mode "Positioning with MDI".
- Rotation** To rotate the coordinate system, you only have to enter the rotation angle H.



- Planes**
- XY plane:** G17 +X axis = 0° (standard)
- YZ plane:** G18 +Y axis = 0°
- ZX plane:** G19 +Z axis = 0°
- All coordinate inputs following the rotation are then referenced to the rotated coordinate system.
- The rotation angle is entered in degrees (°).
Input range: -360° to +360° (absolute or incremental).



- Activating the rotation** G73 H+35 *
- The active rotation angle is indicated by "ROT" in the status display.
- Cancelling the rotation** A rotation is cancelled by entering the rotation angle 0°.
- G73 H+0 *

Coordinate Transformations

Coordinate system rotation: G73



Selecting the cycle

Initiate the dialog



ROTATION ANGLE ?

9

0

ENT

Absolute dimensions or

9

1

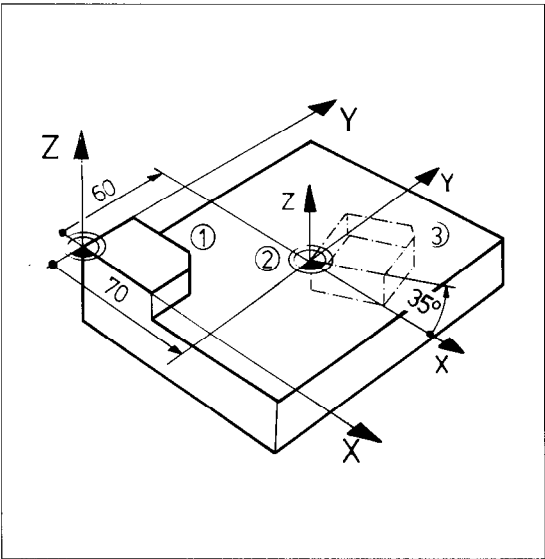
ENT

incremental.Enter rotation angle.

END

Example

A program section (subprogram 1) is to be executed – as originally programmed – at position X+0/Y+0. It is then rotated in X and executed at the position X+70/Y+60.



```
%35 G71 *  
N10 G30 G17 X+0 Y+0 Z-40 *  
N20 G31 G90 X+100 Y+100 Z+0 *
```

```
N30 G99 T1 L+0 R+5 *  
N40 T1 G17 S200 *
```

```
N50 L1,0 *
```

```
N60 G54 X+70 Y+60 *  
N65 G73 H+35 *
```

```
N70 L1,0 *
```

```
N80 G54 X+0 Y+0 *  
N90 G73 H+0 *
```

```
N100 G00 G40 Z+50 M02 *
```

Non-rotated execution ①

Rotated execution. Sequence:

1. Datum shift ②
2. Rotation ③

3. Subprogram call

Cancel datum shift
Reset rotation

Return jump to first block of the main program

Subprogram

The associated subprogram (see "Datum shift") is programmed after M02.

Coordinate Transformations

Scaling: G72



The cycle

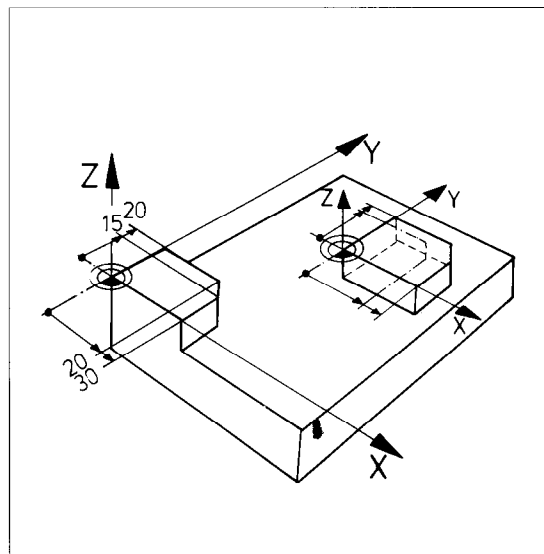
Contours can be enlarged or reduced with this cycle. This permits generation of contours geometrically similar to an original without reprogramming, and also use of shrinkage and growth allowances.



Scaling is effective – depending on the specified machine parameters – either in the machining plane or in the three main axes (see index A, “General Information, MOD Functions, User parameters”).

Activation

Scaling is effective immediately, without being called. Scaling factors greater than 1 result in enlargement, factors between 0 and 1 result in reduction.

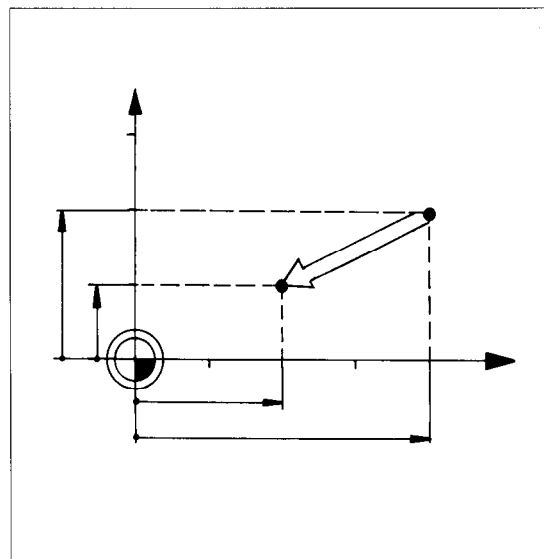


F factor

The scaling factor F (factor) is entered to enlarge or reduce a contour.

The control applies this factor to all coordinates and radii either in the machining plane or (depending on MP 7410; see index A “General Information, MOD Functions, User parameters”) in all three axes X, Y and Z. The factor also affects dimensions in cycles.

Input range: 0.000001 to 99.999999.



Datum position

It is helpful to locate the datum on an edge of the subcontour. This way, the datum of the coordinate system is retained during a reduction or magnification as long as it is not subsequently moved or if the move is programmed before the scaling factor.

Activating scaling

G72 F0.8 *

Cancelling scaling

The scaling cycle is cancelled by entering the factor 1 in the scaling cycle:

G72 F1 *

Coordinate Transformations

Scaling: G72



Selecting the cycle

Initiate the dialog



FACTOR ?

Enter the scaling factor.

Conclude block.

Example

A program section (subprogram 1) is to be executed – as originally programmed – referenced to the manually set datum X+0/Y+0. It is then scaled with 0.8 and executed at the datum X+60/Y+70.

```
%36 G71 *
N10 G30 G17 X+0 Y+0 Z-40 *
N20 G31 G90 X+100 Y+100 Z+0 *
```

```
N30 G99 T1 L+0 R+5 *
N40 T1 G17 S200 *
```

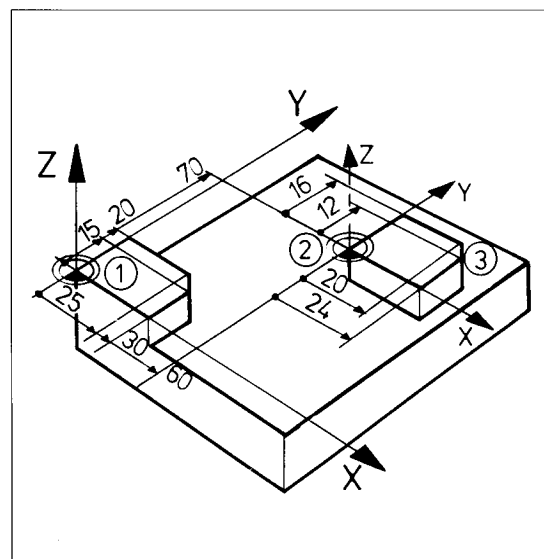
```
N50 L1,0 *
```

```
N60 G54 X+70 Y+60 *
N70 G72 F0,8 *
```

```
N80 L1,0 *
```

```
N90 G54 X+0 Y+0 *
N100 G72 F1 *
```

```
N110 G00 G40 Z+50 M02 *
```



Execution in original size ①

Execution with scaling factor. Sequence:

1. Shift datum ②
2. Define scaling factor ③
3. Call subprogram (scaling factor effective)
Cancel transformations

Retract, return jump

Subprogram

The corresponding subprogram (see cycle 7, Datum shift) is programmed after M02.

Other Cycles

Dwell time: G04



The cycle

In a program which is being run, the next block will be executed only after the end of the programmed dwell time. Modal conditions, such as spindle rotation, are not affected.

Activation

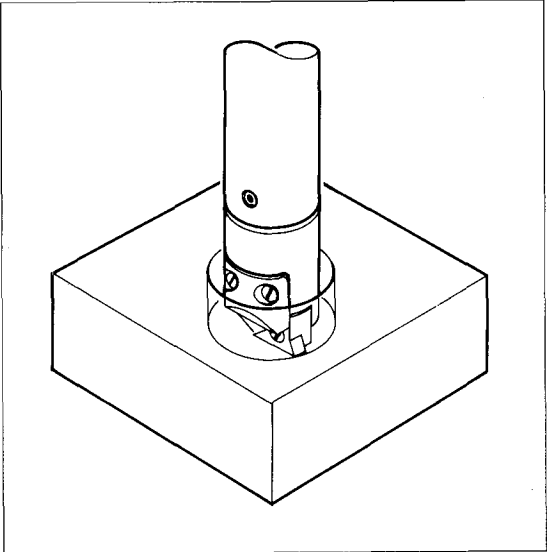
The dwell cycle is active immediately upon definition, without being called.

Possible applications

For example, chip breaking can easily be programmed with a dwell cycle after every drilling step.

Input range

The dwell time is specified in seconds.
Input range: 0 to 30 000 s (≅ 8.3 hours)



Cycle definition

Initiate the dialog



DWELL TIME IN SECS. ?

Enter desired dwell time, in seconds.

Conclude block.

Example

G04 F0.5 *

Other Cycles

Program call: G39



The cycles

Machining procedures that you have programmed – such as special drilling cycles, curve milling, or geometry modules – can be created as callable main programs and be used like fixed cycles. They can be called from any program with a cycle call. They can thus help speed up programming and improve safety, since you are using proven modules.

G39

A callable program defined as a cycle more or less becomes a fixed cycle.

It can be called with

- G79 (separate block) or
- M99 (blockwise) or
- M89 (modally).

Initiate the dialog



Entering the cycle selection

PROGRAM NUMBER ?

Enter program number.

END

Conclude block.

Example

The callable program 50 is to be called from program 5.

Program:
%5 G71 *

G39 P01 50 *

G01 X+20 Y+50 F250 M99 *

N9999 %5 G71 *

Definition:
"Program 50 is a cycle"
Call program 50 with M99

Cross-reference

Drilling with chip breaking

A realistic example of a program call with G39 can be taken from the drilling example (Parameter programming %7445):

- Subprogram 1 is written separately as %7444 (without G98 L1 or G98 L0).
- %7444 now exists as a callable, additional drilling procedure.
This program can remain stored in the control and be called by any other program, e.g. 7445.
- Subprogram 1 is deleted in the main program 7445.
- Instead of L 1,0, write in %7445:
G39 P01 7444, and M99 in a subsequent positioning block.

Other Cycles

Oriented spindle stop: G36



The cycle

The control can address the machine tool spindle as a 6th axis and turn it to a certain angular position.

Application:

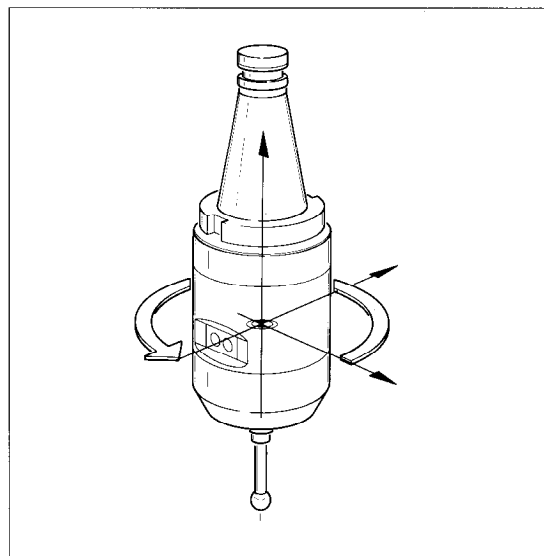
- for tool changing systems with defined change position for tool.
- Orientation of the transmitter/receiver window of the TS 511 3D touch probe system from HEIDENHAIN.

Activation M19

The cycle – if provided on the machine – is executed through **M19**. The spindle orientation is activated either through

- machine parameter or
- spindle orientation: G36

If the cycle is called without prior definition, the spindle will be oriented to the angle set in the machine parameters. Further information is available from the machine tool builder.



Input range

The angle of orientation is entered according to the reference axis of the working plane.
Input range: 0 to 360°.
Input resolution: 0.1°.

Cycle definition

Initiate dialog

G 3 6

ORIENTATION ANGLE ?



Confirm selection of cycle.



Enter new angle for spindle.



Transfer block to memory.

Example

G36 S45 *

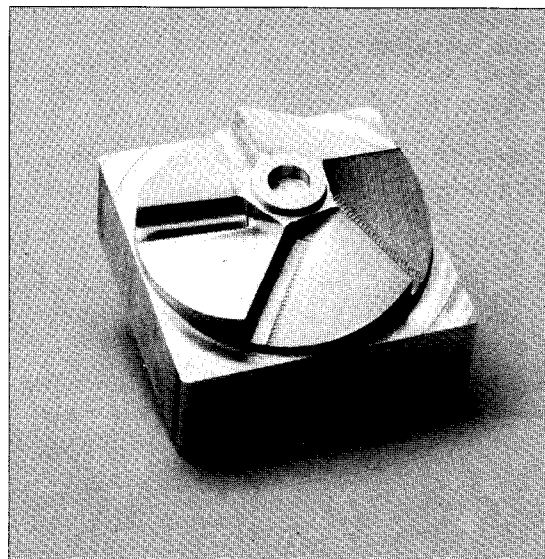
Parametric Programming Overview



Parametric programming

Many problems which would otherwise be impossible or very difficult can be easily solved with parametric programming. Parametric programming expands the capabilities of the control enormously and offers features such as:

- Variable drilling programs
- Processing of mathematical curves (e.g.: sine wave, ellipse, parabola, hyperbola)
- Programs for machining families of parts
- 3D programming for mold making



Basic functions

The mathematical and logical functions listed at the right are available for programming.

Computation time

The time required for one computing step – depending on the workload on the processor – can reach the millisecond range.

For this reason, very many computations and very small displacements may cause the machine axes to be halted. In this case you have to make a compromise between high surface definition (many computations, small displacements) and efficient machining.

- D00: ASSIGN
- D01: ADDITION
- D02: SUBTRACTION
- D03: MULTIPLICATION
- D04: DIVISION
- D05: SQUARE ROOT
- D06: SINE
- D07: COSINE
- D08: ROOT-SUM OF SQUARES
- D09: IF EQUAL, JUMP
- D10: IF UNEQUAL, JUMP
- D11: IF GREATER, JUMP
- D12: IF LESS, JUMP
- D13: ANGLE
- D14: ERROR CODE

Variable addresses with parameters

The program data shown at the right can be kept variable by using the Q parameters:
Enter a Q parameter instead of a specific number.

Example for variable positioning:
instead of X+20.25 you write X+Q21.

The parameter value for Q21 must be computed in the program or be defined before it is called.

Inch dimensions

Programs using parameters as jump address are not to be switched from mm to inches or vice versa, because the contents of the Q parameters are also converted during switchover, which would result in false jump addresses.

Nominal positions G01 X+Q21 Y+Q22 *
Circle data I+Q1 J+Q2 *
G02 X+Q10 Y+Q20 *
G06 X+Q11 Y+Q21 *
G25 Q1 *
G05 X+Q21 Y+Q22 R Q62 *
Feed rate F Q10 *
Tool data G99 T1 L+Q1 R Q2 *
TQ5 G17 SQ6 *
Conditional jump D11 P01+Q10
P02+0 P03 Q30 *
Cycle data G83 P01–Q1
P02–Q2 P03–Q3
P04 Q4 P05 Q5 *

Parametric Programming Selection



**Selecting
basic functions**

D

Basic parameter functions are selected by pressing the “D” key and entering the corresponding number.

**Defining
parameters**

A parameter is designated by the letter Q and any number between 0 and 113. The TNC assigns values to parameters Q100 to Q113.

Specific numerical values (contents) can be allocated to the parameter either directly or with mathematical and logical functions. Parameter contents can also have a negative sign. Positive signs need not be programmed.

**Starting
values**

Parameters must be defined before they can be used. When program run is started, all parameters are automatically assigned the value 0 if machine parameter MP 7300 = 0. If the Q parameters are to be assigned values before program start, set MP 7300 = 1. The Q parameter values are then not deleted at program start.

Examples of defined parameters:
Q1 = +1.5
Q5 = +Q1
Q9 = +Q1 * +Q5

Notation

The notation corresponds to the standard computer format:
The operands and the operator are on the right, the desired result on the left. Consider the entire line as a mathematical operation and not as an equation!

Here also use the “ENT” key to continue the dialog within one program line.

Initiate the dialog

ENT

Q

ENT

END

3

ENT

e. g. multiplication

PARAMETER NUMBER FOR RESULT?		Parameter for result.
First value or parameter?		1 st operand (parameter).
Second value or parameter?		2 nd operand.
		Conclude block.

Example

D03 Q10 P01+Q5 P02+3.142 *

The result is assigned to Q10; the content of Q5 is retained!

Parametric Programming

Algebraic functions



D00: Assignment

This function assigns to a parameter either a numerical value or another parameter.

Example:

D00 Q05 P01 +65.432 *

The assignment corresponds to an equal sign.

D00 Q05 P01 +Q12 *

D00 Q05 P01 -Q13 *

D01: Addition

This function defines a specific parameter to be the sum of two parameters, two numbers or one parameter and one number.

D01 Q17 P01 +Q2 P02 +5 *

D01 Q17 P01 +5 P02 +7 *

D01 Q17 P01 +5 P02 -Q12 *

D01 Q17 P01 -Q4 P02 +Q8 *

D01 Q17 P01 +Q17 P02 +Q17 *

D02: Subtraction

This function defines a specific parameter to be the difference between two parameters, two numbers or one parameter and one number.

D02 Q11 P01 +5 P02 +34 *

D02 Q11 P01 +5 P02 +7 *

D02 Q11 P01 +5 P02 -Q12 *

D02 Q11 P01 +Q4 P02 +Q8 *

D02 Q11 P01 +Q11 P02 -Q11 *

D03: Multiplication

This function defines a specific parameter to be the product of two parameters, two numbers or one parameter and one number.

D03 Q21 P01 +Q1 P02 +60 *

D03 Q21 P01 +5 P02 +7 *

D03 Q21 P01 +5 P02 -Q12 *

D03 Q21 P01 +Q4 P02 -Q8 *

D03 Q21 P01 +Q21 P02 +Q21 *

D04: Division

This function defines a specific parameter to be the quotient of two parameters, two numbers or one parameter and one number.

D04 Q17 P01 +Q2 P02 +62 *

D04 Q17 P01 +5 P02 +7 *

D04 Q17 P01 +5 P02 -Q12 *

D04 Q17 P01 +Q4 P02 +Q8 *

Division by 0 is not permitted!

D05: Square root

This function defines a specific parameter to be the square root of one parameter or one number. The operand must be positive.

D05 Q98 P01 +2 *

D05 Q98 P01 +Q12 *

D05 Q98 P01 -Q70 *

Sign for operands

Parameters with negative signs can also be used.

Q11 = +5 - -Q34

A subtraction can be obtained from an addition and vice versa. This also applies for other operations.

Parametric Programming

Trigonometric functions



Basics of trigonometry

A circle with radius c is divided symmetrically into four quadrants ① to ④ by the two axes X and Y . If the radius c forms the angle α with the X -axis, the two components a and b of the right-angled triangle depend upon angle α .

Defining the trigonometric functions

$$\sin \alpha = \frac{\text{opposite side}}{\text{hypotenuse}} = \frac{a}{c} \quad \text{or } a = c \cdot \sin \alpha$$

$$\cos \alpha = \frac{\text{adjacent side}}{\text{hypotenuse}} = \frac{b}{c} \quad \text{or } b = c \cdot \cos \alpha$$

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha} = \frac{a}{b}$$

Length of one side

According to the Pythagorean theorem:

$$c^2 = a^2 + b^2 \quad \text{or } c = \sqrt{a^2 + b^2}$$

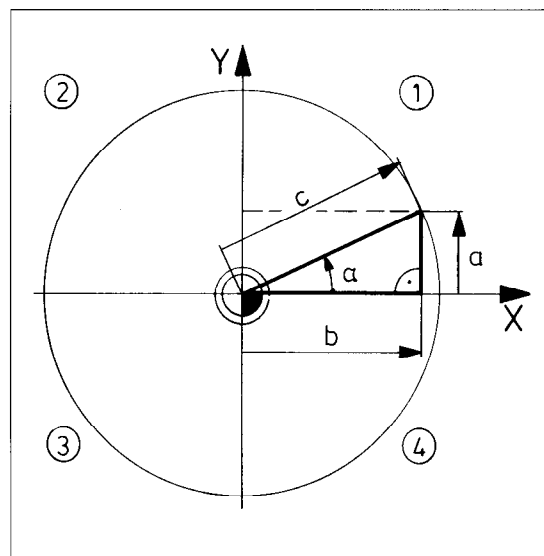


Table for preceding sign and angle range

Function	Quadrant				
	①	②	③	④	
$\sin \alpha$	+	+	-	-	
$\cos \alpha$	+	-	-	+	
$\tan \alpha$	+	-	+	-	
Angle	0°	90°	180°	270°	360°

D06: Sine

A parameter is defined as the **sine** of an angle, whereby the angle can be a number or a parameter (unit of measurement of the angle: degrees).

$$Q44 = \sin Q11$$

$$D06 \text{ Q44 P01 +Q11 *}$$

D07: Cosine

A parameter is defined as the **cosine** of an angle, whereby the angle can be a number or a parameter (unit of measurement of the angle: degrees).

$$Q81 = \cos Q11$$

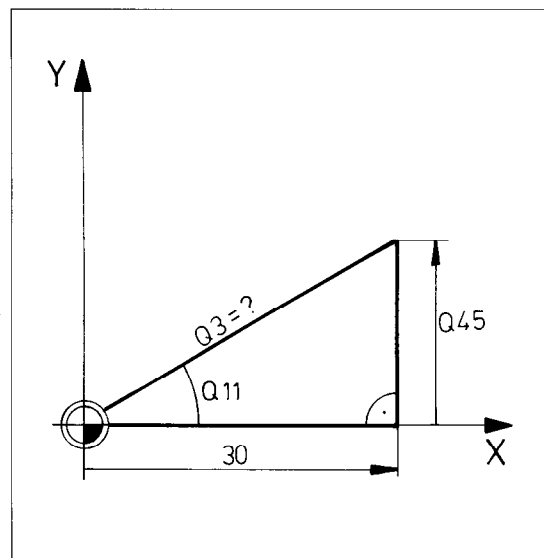
$$D07 \text{ Q81 P01 +Q11 *}$$

D08: Root sum of squares

A parameter is computed as the **square root** of the sum of squares of two numbers or parameters (LEN = length).

$$Q3 = \sqrt{Q45^2 + 30^2}$$

$$D08 \text{ Q3 P01 +Q45 P02 +30 *}$$



Parametric Programming

Trigonometric functions



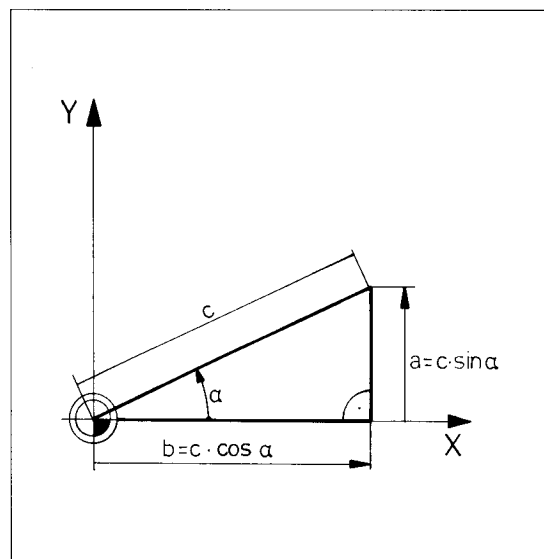
Angles from line segments or trigonometric functions

According to the definitions of the angular functions, either the angular functions $\sin \alpha$ and $\cos \alpha$, or the lengths of sides a and b can be used to determine $\tan \alpha$:

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha} = \frac{a}{b}$$

The angle α is therefore

$$\alpha = \arctan \frac{\sin \alpha}{\cos \alpha} = \arctan \frac{a}{b}$$



Unambiguous angle

If the value of $\sin \alpha$ or the side a is known, two possible angles always result:

Example: $\sin \alpha = 0.5$

$$\alpha_1 = +30^\circ \text{ and } \alpha_2 = +150^\circ$$

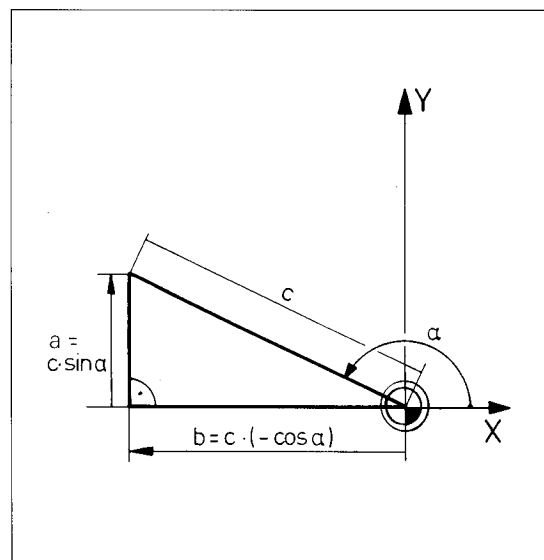
To determine angle α unambiguously, the value for $\cos \alpha$ or side b is required. If this value is known, an **unambiguous** angle α is the result:

Example: $\sin \alpha = 0.5$ and $\cos \alpha = 0.866$

$$\alpha = +30^\circ$$

$\sin \alpha = 0.5$ and $\cos \alpha = -0.866$

$$\alpha = +150^\circ$$



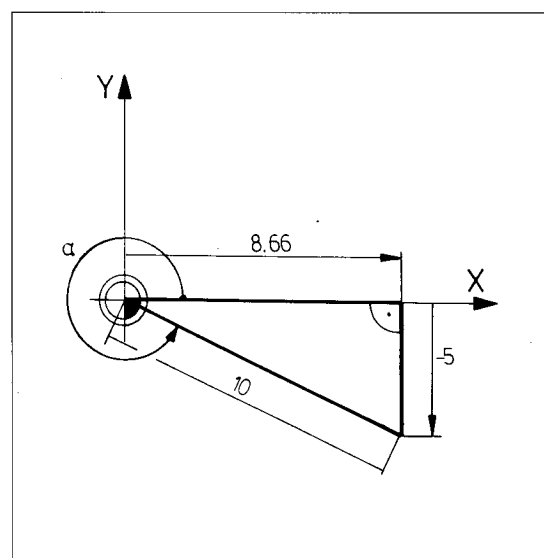
D13: Angle

This function assigns to a parameter the **angle** from a sine and cosine function, or from the two legs of the right-angled triangle.

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha} = \frac{a}{b} = \frac{-5}{8.66}$$

$$\alpha = \arctan \left[\frac{-5}{8.66} \right]$$

D13 Q11 P01 -5 P02 +8.66 *



Parametric Programming

Conditional/unconditional jumps



If-then jump

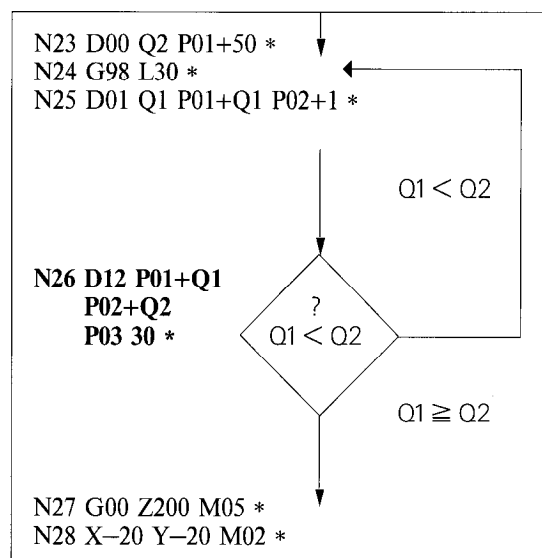
With the parameter functions D09 to D12, you can compare one parameter with another parameter or with a given number (e.g. a maximum value).

Depending on the result of this comparison, a jump to a certain label in the program can be programmed (conditional jump):

If the programmed IF condition is fulfilled, a jump is performed; if the condition is not fulfilled, the next block (following IF ...) will be executed.

Program call

If you write a program call behind the called program label, a jump can be made to another program. (Program calls are for example PGM CALL or cycle G39).



Examples:

Decision criteria:

Equation D09: =

D09 P01+Q1 P02+360 P03 30 *

A parameter is equal to a value or a second parameter, e.g. $Q1 = Q2$ or in the example: Q1 has the value 360.000.

Inequalities D10: ≠

D10 P01+Q1 P02+Q2 P03 2 *

A parameter is not equal to a value or a second parameter, e.g. $Q1 \neq Q2$

D11: >

D11 P01+Q1 P02+360 P03 17 *

A parameter is greater than a value or a second parameter, e.g. $Q1 > Q2$.
Also possible: greater than zero, i.e. positive.

D12: <

D12 P01+Q1 P02+Q2 P03 3 *

A parameter is less than a value or a second parameter, e.g. $Q1 < Q2$.
Also possible: less than zero, i.e. negative.

Unconditional jumps

You can also program **unconditional jumps** to a label with the parameter functions D09 to D12.

Example:

Decision criterion:

D09 P01+0 P02+0 P03 30 *

The condition is **always** fulfilled, i.e. an **unconditional** jump is performed.

Parametric Programming

Special functions



D14: Error code

You can call error messages and dialog texts of the machine tool builder from the PLC EPROM with D14. To call, enter the error code number between 0 and 499.

The error message terminates program run.
The program must be restarted after the error has been corrected.

The messages are allocated as follows:

Error number	Screen display
0 ... 299	ERROR 0 ... ERROR 299
300 ... 399	PLC ERROR 01 ... PLC ERROR 99 (or dialog determined by the machine tool builder).
400 ... 483	DIALOG 1 ... 83 (or dialog determined by the machine tool builder).
484 ... 499	USER PARAMETER 15 ... 0 (or dialog determined by the machine tool builder).

Example:

D14: ERROR = 100

D15: Print

This function outputs current Q-parameter values through the RS-232-C serial interface. You can also enter numerical values between 0 and 499 instead of Q-parameters. These values call error messages and dialog texts which are stored in the PLC EPROM and are allocated as with D14. You can enter combinations of up to six Q parameters and numerical values.

Example:

D15: PRINT Q1/20/Q9/0/Q17/Q33

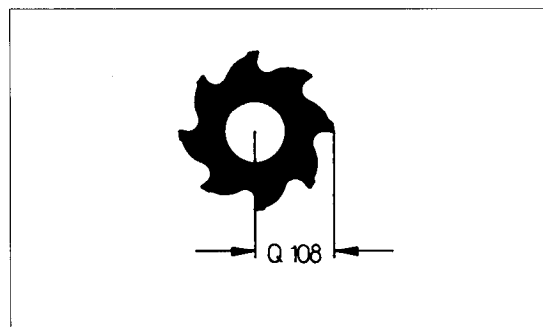
Q100 – Q107

The control can transfer Q parameter values from the integrated PLC to a NC program. The parameters Q100 to Q107 are used for this.

Q108 Tool radius

The control always stores the tool radius of the last called tool in parameter Q108.

The active tool radius can then be used for the radius compensation in parameter computations and comparisons.



Q109 Tool axis

The control stores the current tool axis in parameter Q109:
Different machines alternately use the X,Y or Z axis as the tool axis. On these machines it is helpful when the current tool axis can be requested in the machining program; this makes program branching in user cycles possible.

Current tool axis	Parameter
no tool axis called	Q109 = -1
X axis is called	Q109 = 0
Y axis is called	Q109 = 1
Z axis is called	Q109 = 2

Parametric programming

Special functions



Q110 Spindle on/off

The value in parameter Q110 specifies the last M function issued for the direction of spindle rotation:

M function	Parameter
no M spindle function	Q110 = -1
M03 spindle on clockwise	Q110 = 0
M04 spindle on counterclockwise	Q110 = 1
M05, if M03 was previously issued	Q110 = 2
M05, if M04 was previously issued	Q110 = 3

Q111 Coolant on/off

Parameter Q111 indicates whether the coolant was switched on or off.

Meaning:	Parameter
M08 coolant switched on	Q111 = 1
M09 coolant switched off	Q111 = 0

Q112 Overlap factor

Parameter Q112 contains the overlap factor for pocket milling (see index A "General Information, MOD Functions, User parameters, MP 7430").

The overlap factor for pocket milling can be useful in milling programs.

Q113 mm/inch dimensions

Parameter Q113 specifies whether the NC program at the highest program level (for subprogramming with PGM CALL) contains mm or inch dimensions.

The mainprogram contains:	Parameter
mm dimensions	Q113 = 0
inch dimensions	Q113 = 1

Parametric Programming

Example: Bolt hole circle



Task

A bolt-hole circle is to be drilled using the pecking cycle in the XY plane.

Example:

Radius R of the bolt-hole circle:

Q3 = 35 mm.

Number n of bore holes:

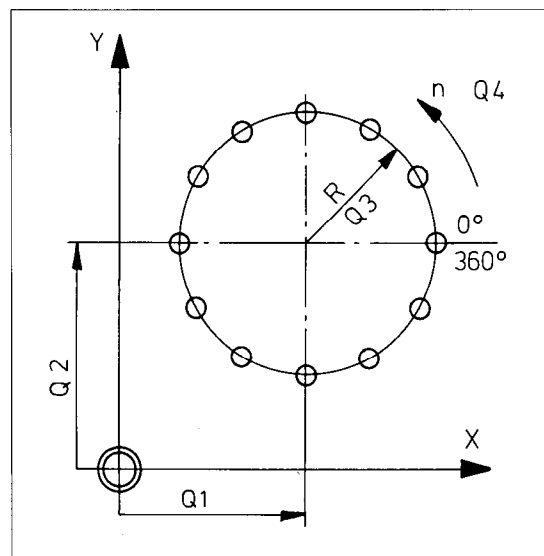
Q4 = 12.

X coordinate of the bolt circle center:

Q1 = 50 mm.

Y coordinate of the bolt circle center:

Q2 = 50 mm.



%37 G71 *

N10 G30 G17 X+0 Y+0 Z-40 *

N20 G31 G90 X+100 Y+100 Z+0 *

N30 G99 T1 L+0 R+5 *

N40 T1 G17 S200 *

Assigning values

N50 D00 Q01 P01 +50 *

N60 D00 Q02 P01 +50 *

N70 D00 Q03 P01 +35 *

N80 D00 Q04 P01 +12 *

N90 G83 P01 -2 P02 -20 *

P03 -5 P04 0 P05 100 *

N100 D00 Q10 P01 +0 *

Computation

N110 D04 Q14 P01 +360 P02 +Q4 *

N120 G00 G90 Z+2 M03 *

Execution

N130 I+Q1 J+Q2 *

N140 G10 R+Q3 H+Q10 M99 *

N150 G98 L1 *

N160 D01 Q10 P01 +Q10 P02 +Q14 *

N170 D09 P01 +Q10 P02 +360 P03 2 *

N180 G10 H+Q10 M99 *

N190 D12 P01 +Q10 P02 +360 P03 1 *

N200 G98 L2 *

N210 G00 G40 Z+50 M02 *

N9999 %37 G71 *

Blank form definition

Define and call tool

Center in X

Center in Y

Bolt circle radius

Number of bore holes

Select and load drilling cycle

Set starting angle

Compute angle increment

Approach setup clearance
and switch on spindle

1st bore

Start of loop

Angle increment

Further bores

If not all holes are drilled, jump to the start of the loop.

Example

Interruptable drilling procedure with automatic approach to the setup clearance and raising of the tool to break the chip for longer tool life.

Main program

```
%7445 G71 *
G30 G17 X+0 Y+0 Z-40 *
G31 G90 X+100 Y+100 Z+0 *
```

D00 Q01 P01 -1 * Setup clearance
 (incremental)

D00 Q02 P01 -40 * Depth (incremental)

D00 Q03 P01 -5 * Infeed (incremental)

D00 Q04 P01 +0,5 * Dwell time

D00 Q05 P01 +200 * Drilling feed rate

D00 Q06 P01 +0 * Work surface
 (absolute)

```
G99 T1 L+0 R+2,5 *    Define tool
T1 G17 S200 *        Call tool,
                          spindle speed
```

```
G00 G90 X+20 Y+50 M03 *
L1,0 *
G00 Z+300 M02 *
```

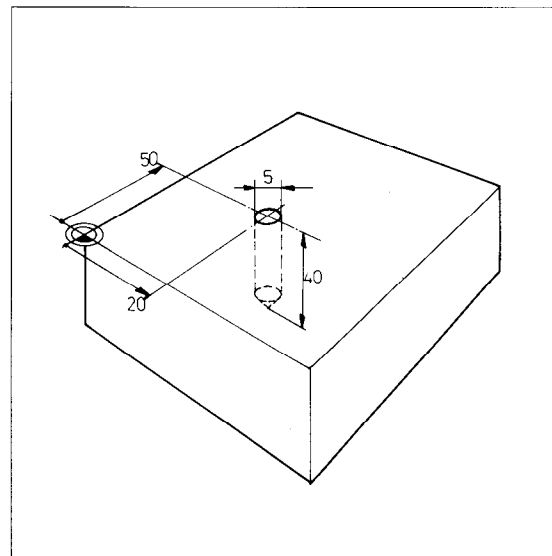
Subprogram 1: Drilling procedure

```
G98 L1 *
D01 Q21 P01 +Q6 P02 -Q1 *
D00 Q23 P01 +Q6 *
D01 Q24 P01 +Q6 P02 +Q2 *
G00 Z+Q21 *
G98 L10 *
D01 Q23 P01 +Q23 P02 +Q3 *
D01 Q22 P01 +Q23 P02 -Q1 *
D12 P01 +Q23 P02 +Q24 P03 99 *
```

```
G01 Z+Q23 FQ5 *
Z+Q22 *
D11 P01 +Q23 P02 +Q24 P03 10 *
```

```
G98 L99 *
G01 Z+Q24 FQ5 *     Drill directly to final depth
G04 FQ4 *           Clear base of bore
G00 Z+Q21 *         Return to setup clearance
G98 L0 *
```

```
%7445 G71 *
```

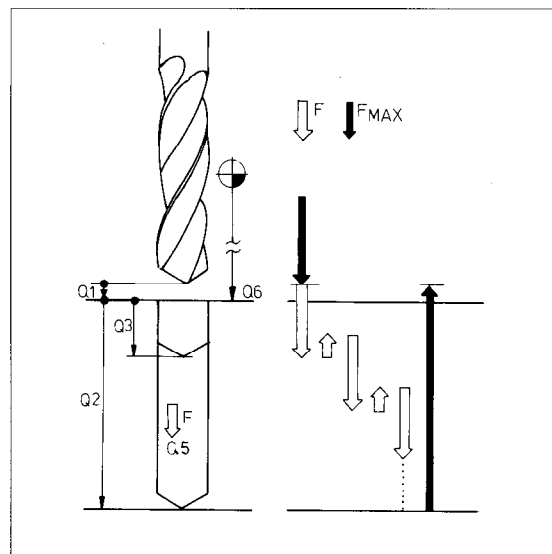


Approach drilling position
Drilling
End of main program

Setup clearance (absolute)
Current work surface (absolute)
Final drilling depth (absolute)
Approach setup clearance in rapid traverse

Compute (new) drilling depth
Compute (new) chip breaking height
Drilling depth would not be attained

Drilling
Chip breaking
Another drilling step required?



Parametric Programming

Example: Ellipse as an SL cycle



Programming of a mathematical curve will be illustrated with an ellipse.

Geometry

An ellipse is defined according to the following formula (parameter form of the ellipse):

$$X = a \cdot \cos \alpha$$

$$Y = b \cdot \sin \alpha$$

a and b are referred to as the semi-axes of the ellipse. Starting at 0° ($Q2$ = starting angle α_a) and increasing α in small increments ($Q1$ = incremental angles $\Delta\alpha$) to 360° ($Q3$ = end angle α_e), a multitude of points on an ellipse results. If these points are connected by short straight lines (see part program below, block N320), a closed contour is produced.

Note

The sine and cosine functions are described in detail under "Parametric Programming, Trigonometric functions".

Process

The machining direction of the ellipse (counter-clockwise) and the selected radius compensation **G41** produce an inside contour (pocket). The contour is contained in the subprogram with program section repeat.

Roughing out

With the SL cycle "contour geometry" (G37), you can write a parameter program as an SL subprogram and execute this with the SL cycle "rough-out" (G57) by selecting an appropriate incremental angle.

Error message

TOO MANY SUBCONTOURS

If the incremental angle ($\Delta\alpha = Q0$) selected for roughing out is too small, the control calculates too many short straight lines, which are interpreted as excessive subcontours.

Remedy

A relatively large incremental angle (e.g. $Q0 = 10^\circ$) suffices for roughing out.

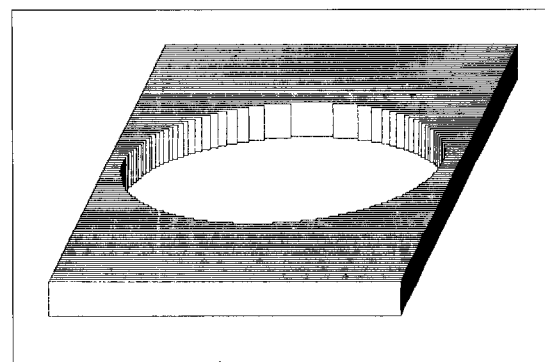
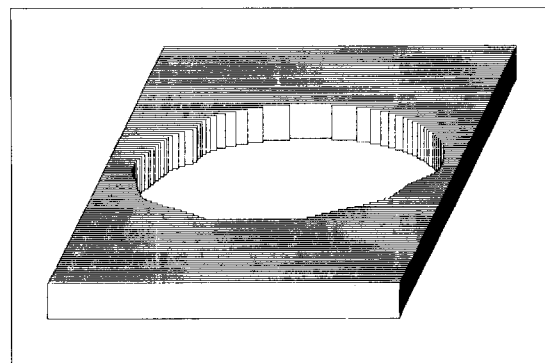
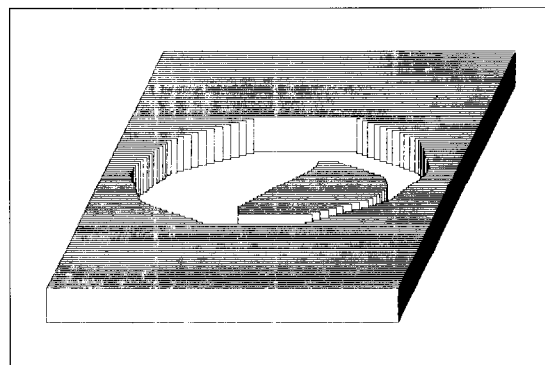
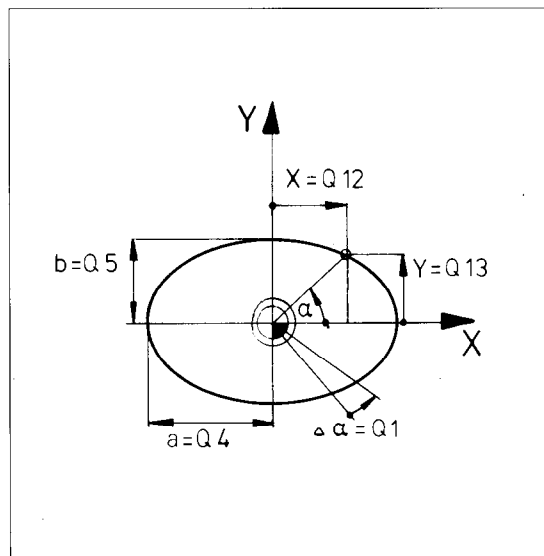
Finishing

For subsequent finishing, the subprogram is executed in the conventional manner with a finer incremental angle (e.g. $Q1 = 1^\circ$).

Note

This program works with only one tool. It can be expanded to use a roughing cutter for "roughing out" (G57) and a finishing cutter for "finishing" (G58/G59).

Also, a center-cut end mill (ISO 1641) is required or cycle G56 is to be applied for pilot drilling.



Parametric Programming

Example: Ellipse as an SL cycle



%94152500 G71 *

Parameter-definition

N10 D00 Q00 P01 +10 *
 N20 D00 Q01 P01 +1 *
 N30 D00 Q02 P01 +0 *
 N40 D00 Q03 P01 +370 *
 N50 D00 Q04 P01 +45 *
 N60 D00 Q05 P01 +25 *
 N70 D00 Q06 P01 +50 *
 N80 D00 Q07 P01 +50*
 N90 D00 Q08 P01 +2 *
 N100 D00 Q09 P01 -5 *

Incremental angle $\Delta\alpha$ for contour roughing
 Incremental angle $\Delta\alpha$ for contour finishing
 Starting angle α_a
 End angle $\alpha_e^{*)}$
 Semiaxis a
 Semiaxis b
 X coordinate for the datum shift
 Y coordinate for the datum shift
 Setup clearance Z
 Pecking depth Z

N110 G30 G17 X+0 Y+0 Z-10 *
 N120 G31 G90 X+100 Y+100 Z+0 *
 N130 G99 T25 L+0 R+2.5 *
 N140 T25 G17 S1000 *
 N150 G00 G40 G90 Z+50 M06 *
 N160 Z+Q8 M03 *
 N170 D00 Q14 P01 +Q2 *
 N180 G54 X+Q6 Y+Q7 *

Blank form definition

N190 G37 P01 2 *
 N200 G57 P01 -Q8 P02 +Q9 P03 -5
 P04 100 P05 +2 P06 +45
 P07 100 *

Copy starting angle for counter
 Datum shift

Define subprogram 2 - as contour label
 SL cycle rough-out
 (for more information, see "SL Cycles")

Roughing out

N210 G79 *

Cycle call

N220 D00 Q00 P01 +Q1 *
 N230 D00 Q14 P01 +Q2 *
 N240 G01 Z+Q9 F100 *
 N250 L2,0 *

Copy incremental angle for finishing
 Copy starting angle for counter
 Drive tool to milling depth Z
 Call subprogram 2

Finishing

N260 Z+50 F1000 M02 *

Retract spindle axis, jump to
 start of program

Subprogram with program section repeat

N270 G98 L2 *
 N280 D07 Q10 P01 +Q2 *
 N290 D06 Q11 P01 +Q2 *
 N300 D03 Q12 P01 +Q10 P02 +Q4 *
 N310 D03 Q13 P01 +Q11 P02 +Q5 *
 N320 G01 G41 X+Q12 Y+Q13 F200 *
 N330 D01 Q02 P01 +Q2 P02 +Q0 *
 N340 D12 P01 +Q2 P02 +Q3 P03 2 *
 N350 G98 L0 *

Label 2
 Computation of the X and Y positions on the
 elliptical path

Feed rate for finishing
 Increase angle
 If angle not attained, jump to label 2

N9999 %94152500 G71 *

*) End angle α_e is greater than 360°, so the contour is safely completed with the cutter.

Modified program

If only the curve of the ellipse is to be milled, lines N10 and N190 to N210 are not needed.
 Line N240 (drive tool to milling depth Z) is inserted behind line N320.

Parametric Programming

Example: Sphere



Task Program 7513 machines a convex segment of a sphere using concentric circular movements in the horizontal plane.

Geometry The size and location of the sphere can be entered.

You obtain a hemisphere when you select:

Starting 3D angle Q1 = 0°
End 3D angle Q2 = 90°
Starting plane angle Q6 = 0°
End plane angle Q7 = 360°

Cutting conditions Cutting is performed during both the advance and return movements.
The following can be selected:
3D angle increment Q3
Downfeed rate Q11
Milling feed rate Q12

Note When selecting the 3D incremental angle, you have to make a compromise between the desired surface quality and the machining time. Small 3D incremental angles must be selected for high surface quality, but they require correspondingly long machining times.

Tool required A spherical cutter is used for finishing.

Assigning values

```
%7816 G71 *  
N10 D00 Q01 P01 +10 * Starting 3D angle  
N20 D00 Q02 P01 +55 * End 3D angle  
N30 D00 Q03 P01 +1 * 3D incremental angle  
N40 D00 Q04 P01 +40 * Sphere radius  
N50 D00 Q05 P01 +45 * Setup clearance in Z  
N60 D00 Q06 P01 -90 * Starting plane angle  
N70 D00 Q07 P01 +90 * End plane angle  
N80 D00 Q08 P01 +50 * X sphere center  
N90 D00 Q09 P01 +50 * Y sphere center  
N100 D00 Q10 P01 -40 * Z sphere center  
N110 D00 Q11 P01 +100 * Downfeed rate  
N120 D00 Q12 P01 +500 * Milling feed rate
```

Blank

```
N130 G30 G17 X+0 Y+0 Z-50 *  
N140 G31 G90 X+100 Y+100 Z+0 *
```

Tool

```
N150 G99 T1 L+0 R+5 *  
N160 T0 G17 *
```

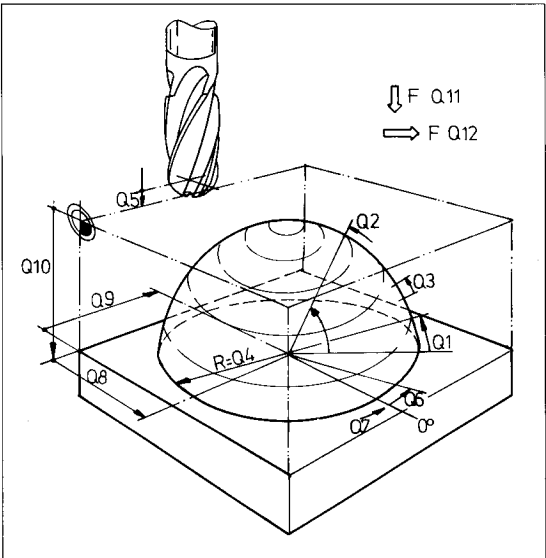
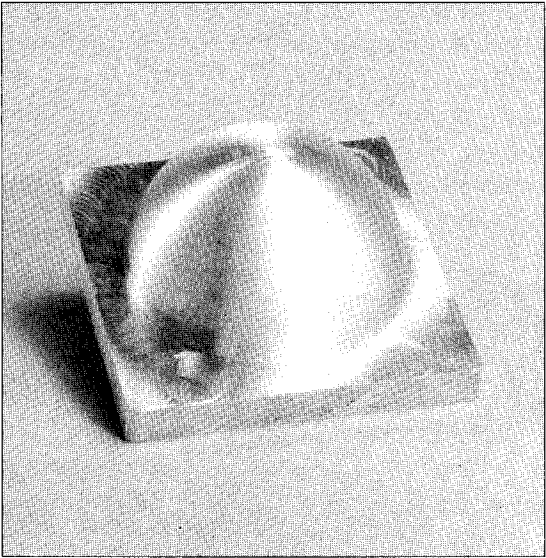
Change/Start position

```
N170 G00 G90 Z+100 M06 *  
N180 T1 G17 S800 *
```

Subprogram call N190 L2,0 *

N200 G00 Z+100 M02 *

Roughing If roughing is required, an end mill can be used with a correspondingly larger sphere radius (Q4).



Parametric Programming

Example: Sphere



Setting the starting values

```
N210 G98 L2 *
N220 G54 X+Q8 Y+Q9 Z+Q10 *
N230 I+0 J+0 *
N240 D00 Q20 P01 +Q1 *
N250 D01 Q31 P01 +Q4 P02 +Q108 *
N260 L3,0 *
N270 G10 G40 R+Q17 H+Q6 M03 *
N280 G00 Z+Q5 *
N290 G01 Z+Q15 FQ11 *
N300 G13 H+Q7 FQ12 *
```

Move datum to the sphere center
Set circle center
Starting and current 3D angle
Compensate sphere radius (with tool radius)
Compute starting position
Approach starting position
Approach setup clearance
Plunge cut at downfeed rate
Circle segment to plane end angle

Starting position

Program loop

```
N310 G98 L1 *
N320 D01 Q20 P01 +Q20 P02 +Q3 *
N330 D11 P01 +Q20 P02 +Q2 P03 99 *
N340 L3,0 *
N350 G01 Z+Q15 FQ11 *
N360 G11 R+Q17 FQ12 *
N370 G12 H+Q6 *
N380 D01 Q20 P01 +Q20 P02 +Q3 *
N390 D11 P01 +Q20 P02 +Q2 P03 99 *
N400 L3,0 *
N410 G01 Z+Q15 FQ11 *
N420 G11 R+Q17 FQ12 *
N430 G13 H+Q7 FQ12 *
N440 D12 P01 +Q20 P02 +Q2 P03 1 *
```

3D angle increment
If condition* is fulfilled, then jump to end
Position computation
Pre-positioning for withdrawal

Return to plane starting angle
3D angle increment
If condition* is fulfilled, then jump to end
Position computation
Pre-positioning

Arc to plane end angle
If condition* is fulfilled, then jump to start of loop

End

```
N450 G98 L99 *
N460 G00 Z+Q5 *
  
N470 G54 X+0 Y+0 Z+0 *
N480 G98 L0 *
```

Finished, retract
Reset datum

Position computations

```
N490 G98 L3 *
N500 D06 Q14 P01 +Q20 *
N510 D03 Q15 P01 +Q14 P02 +Q31 *
N520 D07 Q16 P01 +Q20 *
N530 D03 Q17 P01 +Q16 P02 +Q31 *
N540 G98 L0 *
```

Computations
Z components

Radius components

* Condition: if current 3D angle Q20 is greater than or less than end 3D angle Q2, then jump to ...

Computation values

Q15: Current Z height
Q17: Current radius (polar radius)
Q20: Current 3D angle
Q31: Compensated contour radius
Q108: Current tool radius

Cycle sphere

The program can be used as a cycle:

1. Subprogram 2 (blocks N210 to N480) is written as a separate program.
2. Lines N210 and N480 are not required. Subprogram 3 (blocks N490 to N540) is written in place of block N260.
3. The user need only write the surrounding program (blocks N10 to N200) and then call the cycle in block N190 (PGM CALL).

Parametric Programming

Example: Sphere



Machining sections of a hemisphere

Program %7816 can also be used to machine sections of a hemisphere by limiting the plane angles and 3D angles.

The graphic always shows the surface as cut by a cylindrical end mill.

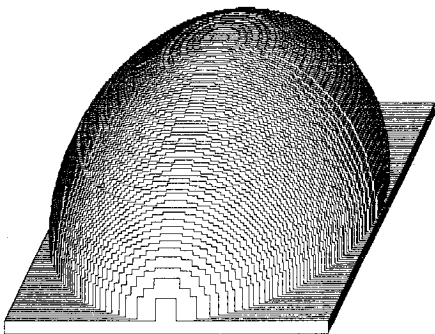
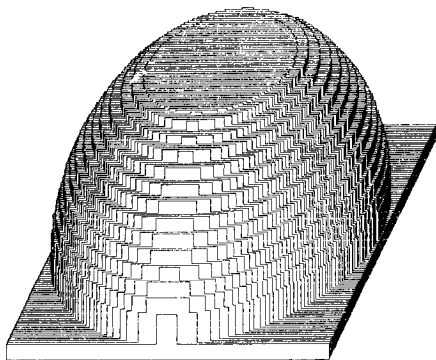
Roughing

End mill, $R = 12\text{ mm}$,
3D angle increment 4°

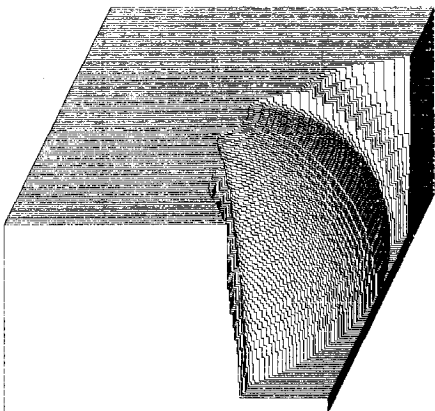
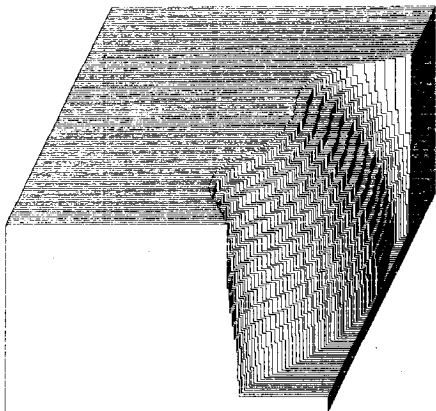
Finishing

Spherical cutter, $R = 3\text{ mm}$,
3D angle increment 1°

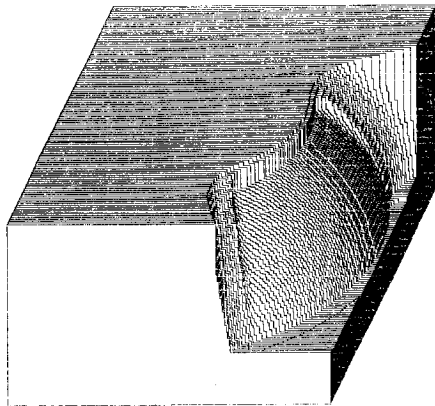
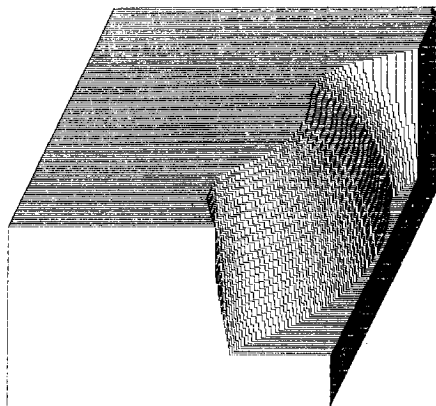
Hemisphere:
3D angle
 0° to 90°
Plane angle
 0° to 360°



3D angle
 0° to 90°
Plane angle
 -60° to 20°



3D angle
 10° to 55°
Plane angle
 -60° to 20°

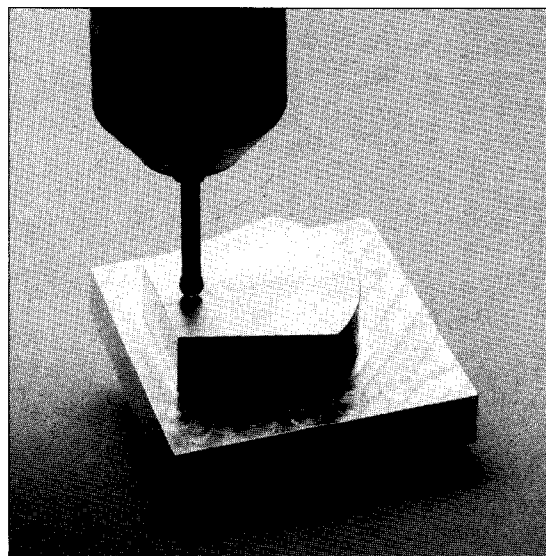


Programmed Probing Overview



The programmable probing function enables you to take dimension measurements before or during a program run. You can probe the upper surfaces of castings with varying heights, for example, to ensure that each is machined to the proper depth.

In addition, thermally-induced position deviations of the machine can be determined at selected time intervals and compensated.



Process

The probe moves to the starting position while maintaining the setup clearance (machine parameter). It then approaches the workpiece at the measuring feed rate. Upon contact, the probed position is stored and the probe retracts at rapid traverse to the setup clearance.

If the stylus does not make contact before reaching the maximum probing depth (machine parameter), the operating is aborted.

Input

Initiate the dialog



PARAMETER NUMBER FOR RESULT

☐ Parameter number

PROBING AXIS/PROBING DIRECTION ?

☒ Probing axis and probing direction

☒ ☐ All coordinates of the starting position, incremental, if desired

☒ ☐

☒ ☐



Conclude block

Example

The probe is first to be pre-positioned to X-10, Y+20 and Z-20, and then probing begun with the X axis in positive direction. The probed result (X position) is to be stored in Q10.

Program

T0 G17 *

G00 G40 Z+200 M06 *

G55 P01 10 P02 X+

G90 X-10 Y+20 Z-20 *

Tool change position

Probing with the X axis in positive direction, measuring result in Q10

Pre-positioning

Q10 contains the **compensated** X axis measurement after probing

Programmed Probing: G55

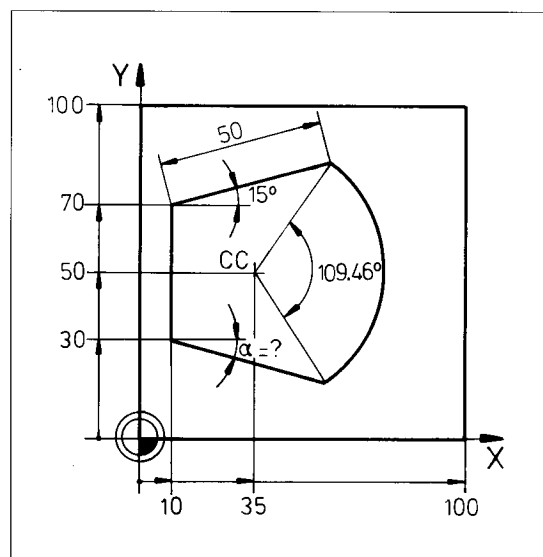
Example: Measuring length and angle



Task A length (from the probing points ① and ②) and an angle (from the probing points ③ and ④) are to be measured with parameter programming.

Note The following program is a solution to the drawing at right.

The theory behind the measurement of angles is explained briefly in "Parameter Programming, Trigonometry functions".



Main program: Definition of probing points (pre-positioning)

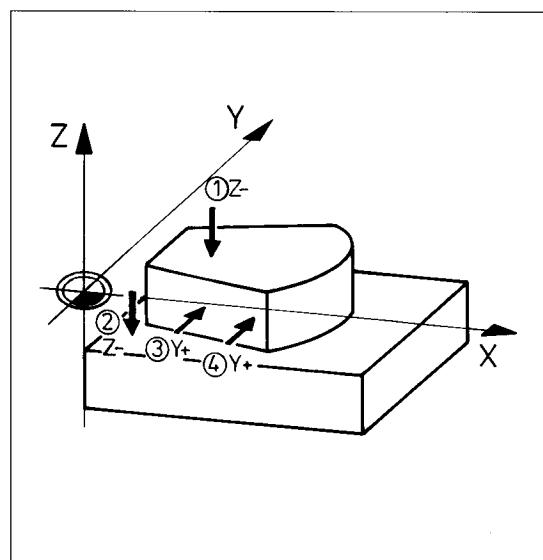
```
%129 G71 *
N10 D00 Q11 P01 +20 *   Probing point ①
N20 D00 Q12 P01 +50 *   X, Y, Z coordinates for
N30 D00 Q13 P01 +10 *   pre-positioning

N40 D00 Q21 P01 +20 *   Probing point ②
N50 D00 Q22 P01 +15 *
N60 D00 Q23 P01 +0 *

N70 D00 Q31 P01 +20 *   Probing point ③
N80 D00 Q32 P01 +15 *   Z coordinate Q33 valid
N90 D00 Q33 P01 -10 *   for probing point ④

N100 D00 Q41 P01 +50 *   Probing point ④
N110 D00 Q42 P01 +10 *

N120 T0 G17 *
N130 G01 G90 Z+100 F1000 M06 *   Retract,
                                   insert probe
                                   system
```



Measure length

```
N140 G55 P01 10 P02 Z-
      X+Q11 Y+Q12 Z+Q13 *
N150 G55 P01 20 P02 Z-
      X+Q21 Y+Q22 Z+Q23 *
N160 L1,0 *
```

Measure angle

```
N170 G55 P01 30 P02 Y+
      X+Q31 Y+Q32 Z+Q33 *
N180 G55 P01 40 P02 Y-
      X+Q41 Y+Q42 Z+Q33 *
N190 L2,0 *

N200 G38 *

N210 Z+100 M02*
```

① Probe

Approach auxiliary point
② Probe
Call subprogram 1

③ Probe

④ Probe

Call subprogram 2

Program STOP
Check result parameter (see index M "Machine
Operating Modes, Program run:
Checking/Changing Q Parameters")
Retract, jump to start of program

Programmed Probing: G55

Example: Measuring length and angle



Subprogram 1:
measure length

N260 G98 L1 *
N270 D02 Q01 P01 +Q20 P02 +Q10 *
N280 G98 L0 *
N285 *

Measured **height** or depth Z
in parameter **Q1**.

Subprogram 2:
measure angle

N290 G98 L2
N300 D02 Q34 P01 +Q40 P02 +Q30 *
N310 D02 Q35 P01 +Q41 P02 +Q31 *
N320 D13 Q02 P01 +Q34 P02 +Q35 *
N330 D01 Q02 P01 -360 P02 +Q2 *
N340 G98 L0 *
N9999 %129 G71 *

Measured **angle** in parameter **Q2**.



Tool compensation

Position values (coordinates) acquired via “Capture actual position” contain the length and radius of compensation for the tool in use.

Therefore it is advisable when programming with “Capture actual position” to enter the correct radius compensation (G41, G42 or G43, G44) and use $L = 0$ and $R = 0$ in the tool definition.

If the tool breaks or another tool is selected instead of the original, then a different length and radius can be taken into account.

The new compensation values for the tool radius are differences:

Radius compensation

$$R = R_2 - R_1 \text{ or}$$
$$R = R_3 - R_1$$

R = radius compensation for the tool definition
 R_1 = tool radius of the original tool ①
 R_2 = tool radius of a new tool ②
 R_3 = tool radius of a new tool ③

The compensation R can be **positive or negative**, depending on whether the tool radius of the newly inserted tool is larger (+) or smaller (–) than the original tool.

Length compensation

The compensation for the new tool length is also determined as the difference to the originally used tool (see “Tool definition, Transferring tool length”).

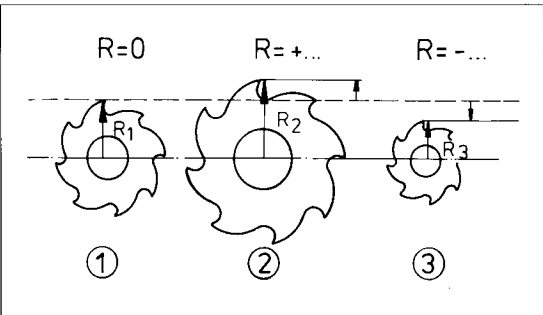
The new compensations are entered in the tool definition of the original tool ($R = 0$, $L = 0$).

1. Tool definition in the part program

3 N10 G95 T1 L+0 R+0

2. Tool definition in the central tool file

T1 L+0 R+0





Capture actual position

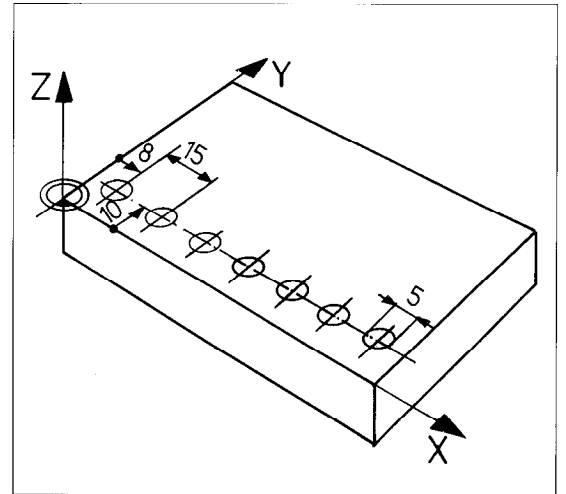


The actual tool position can be transferred to the part program with the "Capture actual position" key.

Applications

In this way you can capture:

- positions
- tool dimensions (see "Tool Definition").



Process

Move the tool to the desired position.

Open a program block (e.g. for a straight line) in the "Programming and editing" operating mode. Select the axis from which the actual value is to be transferred.

This axis position is transferred to memory by pressing the "Capture actual position" key.

PROGRAMMING AND EDITING

```
N25 G00 G40 G90 X+10 Y+10 M03 *  
N30 G54 X+100 Y _____ *  
N40 G28 X *  
N50 I+100 J+0 *  
N60 G73 G90 H+315 *  
N70 G72 F0,8 *  
N9999 X7410 G71 *
```

```
-----  
ACTL. X + 9,375 Y + 8,200  
Z + 8,985 R + 0,180
```

```
T F 0 MS/9
```

Example

X Y Z Move the axis or axes via the axis keys.



Enter radius compensation if required.



Capture positions axis by axis.



Enter feed rate if required.



Enter miscellaneous function if required.



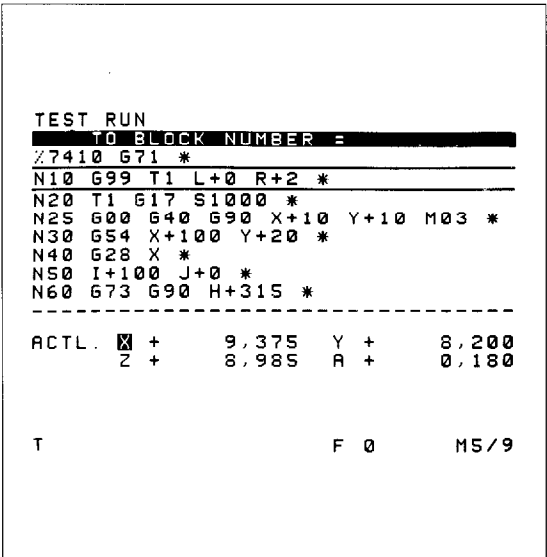
Conclude block.

Input



In the “Test run” operating mode, a machining program is checked for the following errors without machine movement:

- Overrunning the traversing range of the machine
- Exceeding the spindle speed range
- Illogical entries, e.g. redundant input of one axis
- Failure to comply with elementary programming rules e.g. cycle call without a cycle definition
- Certain geometrical incompatibilities



Testing the program

Initiate the dialog



PROGRAM SELECTION

PROGRAM NUMBER =

☐ Select the program to be tested.

TO BLOCK NUMBER =

☐ Key in and confirm the block number up to which the test is to run.

or

☐ Test the complete program.

No apparent errors

If the program contains no apparent errors, the program test runs until the entered block number is reached, or a jump is made back to the start of program if no G38 (STOP) or M06 was programmed.

G38/M06

If a G38 or M06 was programmed, the test can be continued by entering a new block number or by pressing the “NO ENT” key.

Error

If an error is found, the program test is stopped. The error is usually located in or before the stopped block. An error message is displayed on the screen.



The program test can be halted with the “DEL ” key and aborted at any time.

Graphic Simulation



GRAPHICS



Machining programs can be simulated graphically and checked in the "Program run" operating modes "Full sequence" and "Single block", if a blank has been previously defined (G30/G31).

More information on the definition of the blank can be found in the section "Program Selection, Blank form definition".



GRAPHICS

After selecting a program, the menu shown at the right is displayed by pressing the GRAPHICS "MOD" key twice.



One of the versions of the graphic presentations can be selected with the vertical cursor keys and entered with the "ENT" key.



The graphic simulation or internal computation is started with the "START" key.

Fast data image processing

With "Fast data image processing" only the current block number is displayed on the screen and the internal computing also indicated by an asterisk (* = control is started).

When the program has been processed, the "machined" workpiece can be displayed in plan view, view in three planes or 3D view.

Plan view with depth indication

The workpiece center is shown in the plan view with up to 7 different shades: the lower the darker.

View in three planes



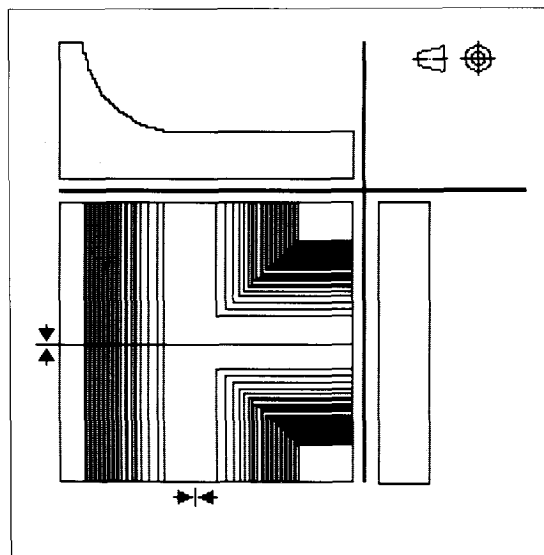
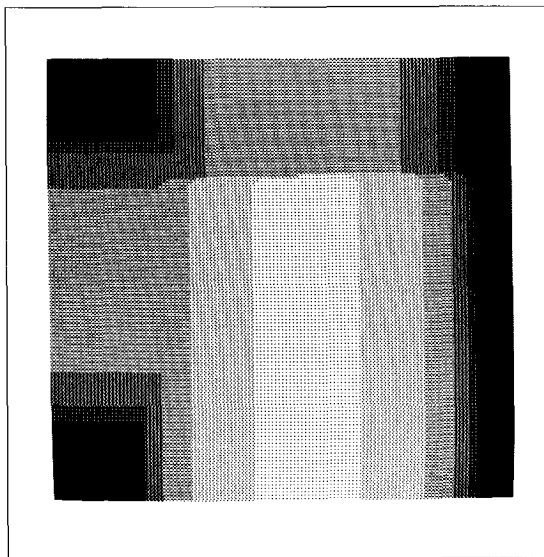
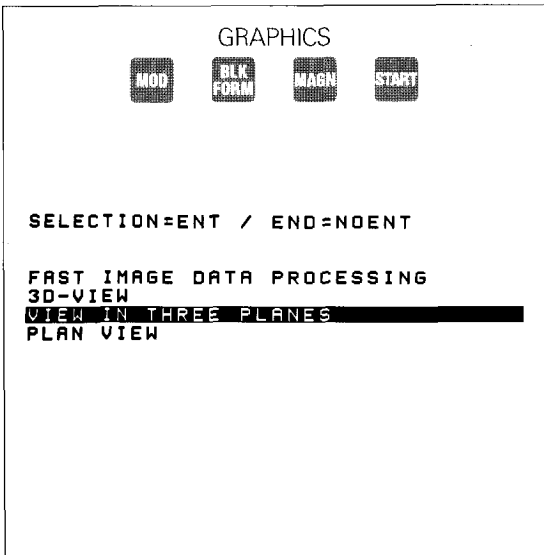
The workpiece is shown – like in drafting – with a plan view and two sections.

The sectional planes can be moved via the cursor keys.

The view in three planes can be switched from the German to the American projection via a machine parameter. A symbol (in conformance to ISO 6433) indicates the type of projection:

Preferred German

Preferred American



Graphic Simulation



GRAPHICS

3D view

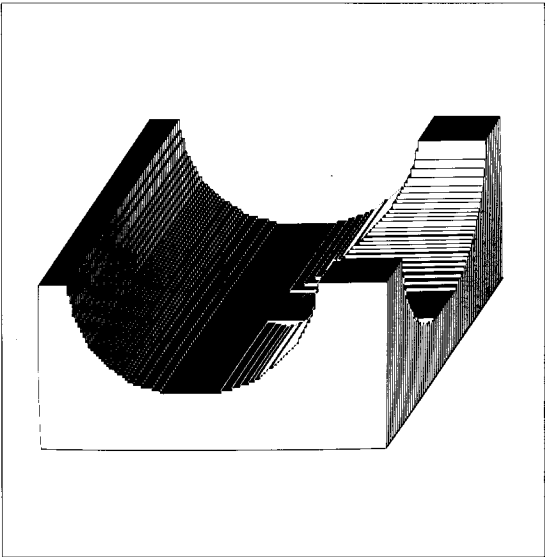
The program is simulated in a three-dimensional view.



The displayed workpiece can be rotated by 90° with each activation of the horizontal cursor keys. The orientation is indicated by an angle.
└ = 0° ┘ = 180°
└ = 90° ┘ = 270°



If the height to side proportion is between 0.5 and 50, the type of display can be changed with the vertical cursor keys. You can switch between a scaled and non-scaled view. The short height or side is shown with a better resolution in the nonscaled view.



Magnifying



You can magnify a detail of the displayed work with the "MAGN" key. A wire model with a hatched surface appears next to the graphic. This marks the sectional plane.

Selecting the sectional plane

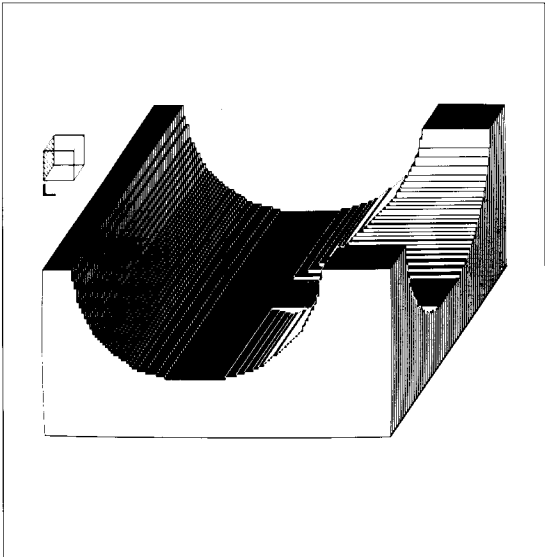


You can select a different sectional plane with the vertical cursor keys.

Trimming



You can trim the selected plane or cancel the section with the horizontal cursor keys.



Magnifying the detail



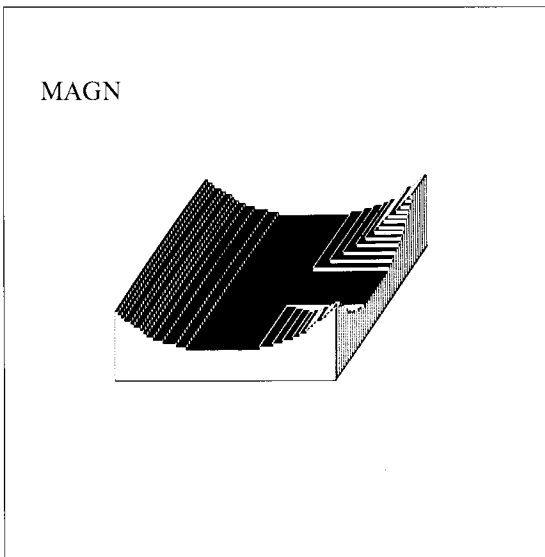
Once the desired detail is displayed, select the dialog "TRANSFER DETAIL = ENT" with the vertical cursor keys and confirm with the "ENT" key.

Magnification



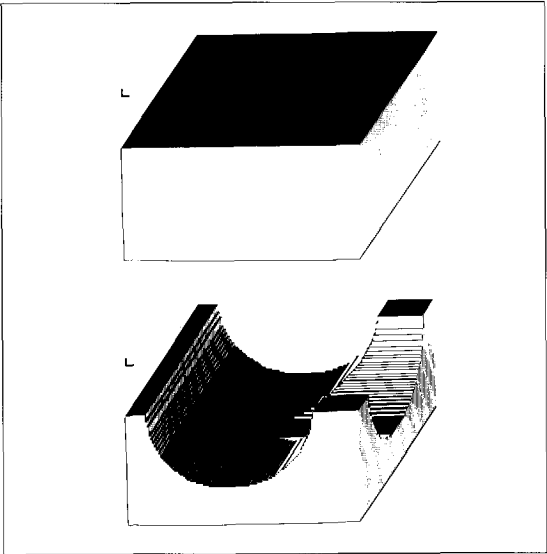
The "remaining workpiece" is displayed on the screen with "MAGN".

Another graphic simulation of machining of the magnified detail can be executed in the plan view, the view in three planes or the 3D view via the "START" key.





You can restore the complete blank with the “BLK FORM” key and restart simulation with “START”.



Tips

The “3D view” and “View in three planes” are especially realistic, but they require extensive computing. For long programs, we therefore recommend displaying the workpiece with “Fast data image processing” or in the quicker “Plan view with depth indication” first, and then switching to the “3D view” or the “View in three planes”.

Displaying details

The following aids are available if fine details are to be examined:

- Trim the blank and magnify in an additional graphic program run.
- Restrict the blank detail to the section of interest.

Tool call



A tool call must be programmed with “T” prior to the first axis movement to designate the tool axis. Specifying the spindle axis in the BLK FORM definition does not suffice for the graphic program run. Both entries for the axis must be the same.

If the tool axis is not given, an error message appears after starting the graphics.

External Data Transfer

General information



The control has one data interface for read-in or output of programs. The data format complies with the following standard:

- RS-232-C (ISO)

The data interface can function in two different manners:

Blockwise transfer for
the HEIDENHAIN FE 401 Floppy Disk Unit and compatible computers.

Standard data transfer for
the HEIDENHAIN ME magnetic tape unit*, or for a printer, punch, reader, etc.

* (no longer in production).

Device adaptation

The TNC can be adapted to various peripheral devices via machine parameters, which can be accessed as user parameters.

The settings for three different peripheral devices are permanently stored in the TNC (selectable via "MOD"):

- FE = for HEIDENHAIN FE 401 Floppy Disk Unit.
- ME = for HEIDENHAIN ME magnetic tape unit.
- EXT = external devices.
Interface defined by the machine manufacturer or user via machine parameters to connect a non-HEIDENHAIN device such as a printer, computer, etc.

External programming

Programs can also be written externally.

Observe the programming rules in this manual and the following instructions.

- At the start of program and after every program block, CR LF or LF or CR FF or FF must be programmed.¹⁾
- After the end of program block, CR LF or LF or CR FF or FF¹⁾ and also ETX (Control C) must be programmed.
Any character can be substituted for ETX.
- Spaces between single words can be omitted.
- Trailing zeros can be omitted.
- During read-in of NC programs, comments that are marked with "*" or ";" are ignored.

¹⁾ CR, LF at the start of program and CR, LF or LF or FF after every block are not required for "blockwise transfer". This function is assumed by the control characters.

External Data Transfer

Transfer menu



Read-in/ read-out



Part programs can be read-out or read-in by the control. For example, the "Read-in program" display on the control means: data is entered from the floppy disk station and received by the control. Program transfer in the "Programming and editing" operating mode must be initiated from the control.

Transfer menu

The transfer mode is selected via a menu, which offers different read-in and read-out alternatives.

PROGRAMMING AND EDITING
SELECTION = ENT/END = NOENT

PROGRAM DIRECTORY
READ-IN ALL PROGRAMS
READ-IN PROGRAM OFFERED
READ-IN SELECTED PROGRAM
READ-OUT SELECTED PROGRAM
READ-OUT ALL PROGRAMS

Selections



Read-in to the TNC

PROGRAM DIRECTORY
The list of program numbers on the data medium is displayed. The programs are not transferred.

READ-IN ALL PROGRAMS
All programs are read-in from the data medium.

READ-IN PROGRAM OFFERED
The programs are offered in the sequence in which they were externally stored and, if desired, can be read-in.

READ-IN SELECTED PROGRAM
A single, selected program is read-in.

Read-out from the TNC

READ-OUT SELECTED PROGRAM
A single, selected program is read-out.

READ-OUT ALL PROGRAMS
The entire NC program memory is read-out.

Interrupting the data transfer

A started data transfer can be interrupted on the TNC by pressing the "END  " key. After interruption of the data transfer, the following error message appears:

PROGRAM INCOMPLETE

Transfer TNC – TNC

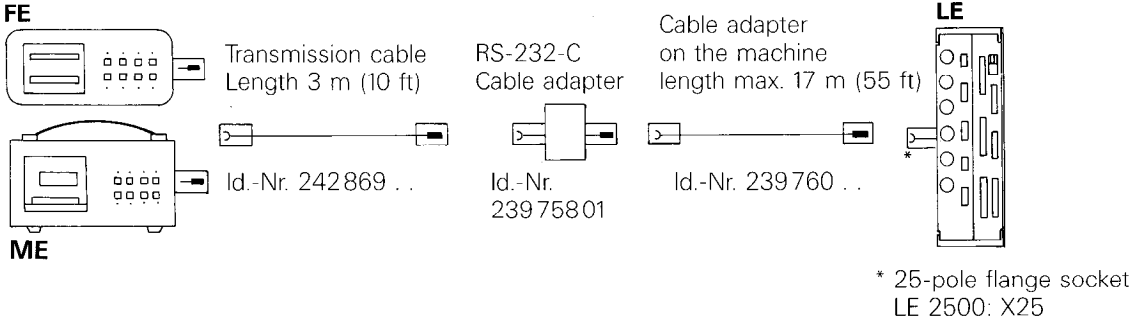
Data can also be transferred directly between two controls. The receiving control must be started first.

External Data Transfer

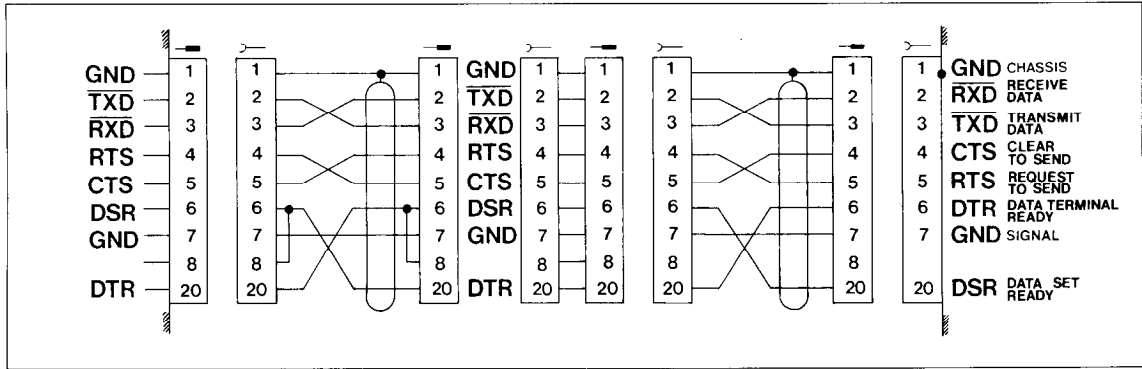
Connecting cable/Pin assignment for RS-232-C



HEIDENHAIN devices

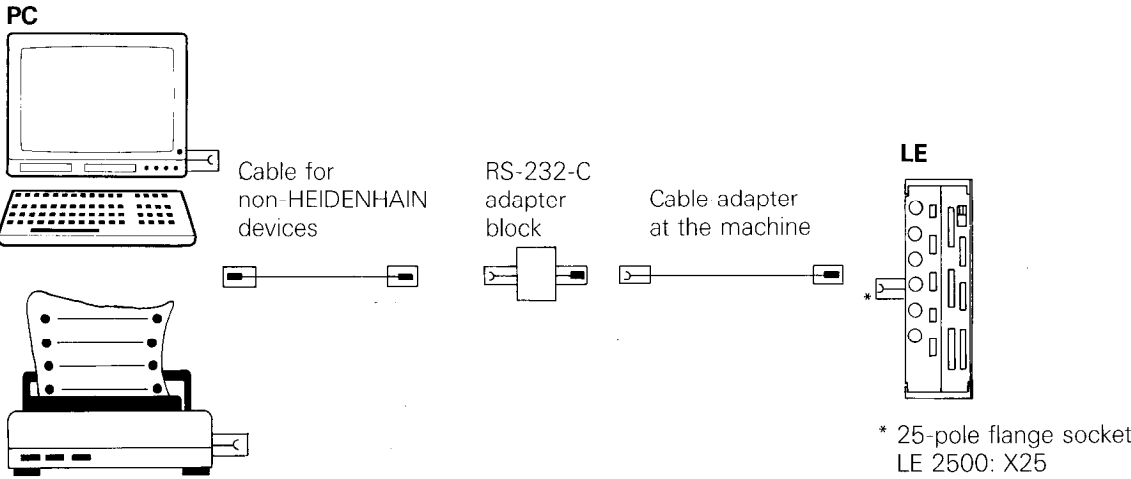


HEIDENHAIN-standard cable

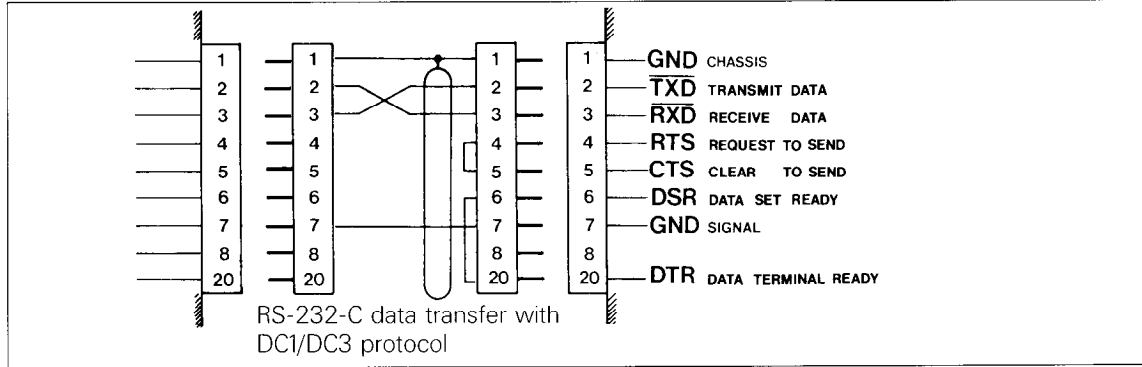


The RS-232-C data interface has a different pin layout at the LE and at the adapter block.

Non-HEIDENHAIN devices



Recommended pin layout for non-HEIDENHAIN devices



External Data Transfer

Peripheral devices



Adaptation	The control interface must be set for the specific device which is to be connected.
HEIDENHAIN devices	HEIDENHAIN devices are mated with the TNC controls and are therefore especially easy to put into operation:
FE, ME	The adaptation for FE or ME can be selected via "MOD". The suitable standard cable can be ordered. The transfer rate can be altered for the FE 401B.
Connections	In built-in controls, peripheral devices can usually be connected via a cable adapter on the operating panel or another accessible location on the machine.
Non-HEIDENHAIN devices	Non-HEIDENHAIN devices must be individually adapted. This also includes: ● Adapting the control via machine parameters. These settings are stored after input and are automatically effective by selecting EXT. ● Adapting the peripheral device, e.g. via switches. ● Setting the baud rates for both devices. ● Wiring the data transfer cable. Please remember: Both sides must be set identically. You should always document the settings!

External Data Transfer

FE floppy disk unit

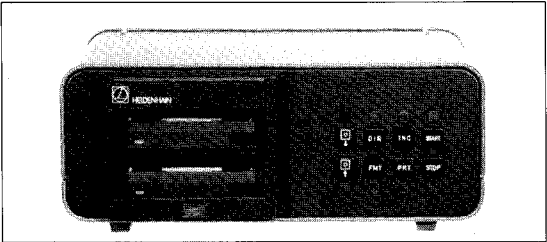


Preparing the FE *)

Connect the FE to the mains, plug in the data cable, switch on, insert floppy disk in the upper drive, select the baud rate if necessary.

Please note when writing a diskette:

- You must format the diskette before writing for the first time.
- Do not write-protect the diskette.



Setting the TNC

Select operating mode at the TNC

Continue pressing until RS-232-C INTERFACE appears.

RS-232-C INTERFACE =

Continue pressing until the FE setting appears.
 Terminate the MOD operating mode.

Examples for using the FE

Read-out selected program

Select operating mode



READ-OUT SELECTED PROGRAM

Confirm the function.

OUTPUT = ENT/END = NOENT

1 13

Select the program, e.g. program 14.

14 24

Output the program.

EXTERNAL DATA OUTPUT

The FE is started and stopped after program transfer.

OUTPUT = ENT/END = NOENT

The next program number is then highlighted.

Select and output the next program or **terminate output. Very important!**



Read-in selected program

Select operating mode



READ-IN SELECTED PROGRAM

Confirm the function.

PROGRAM NUMBER =

Enter the number, read-in.

EXTERNAL DATA INPUT

The FE generally functions with "blockwise transfer" and can be switched over on the rear to operate like an ME.

*) The entire range of functions for the FE is described in the operating manual for the FE.

External Data Transfer

Non-HEIDENHAIN devices



EXT

After setting the TNC data interface to EXT, the following modes can be selected via machine parameter:

Standard data transfer for
printer, reader, puncher etc.

Blockwise transfer for
computer.

To transfer data from the control to non-HEIDENHAIN devices, the control must be adapted by machine parameters.

The transfer rate is set via the MOD function BAUD RATE.

Resetting the TNC to EXT

Select at the TNC



Continue pressing until
RS-232-C interface appears.

RS-232-C INTERFACE



Continue pressing until the EXT setting
appears.



Terminate the MOD operating mode.

Standard data transfer

For standard data transfer (e.g. to a printer), you only have to enter the following machine parameters at the control:

MP 5030 = 0 (standard data transfer is selected).

MP 5020 = e.g. 168 (data format)
(see "External Data Transfer, Machine parameters").

Blockwise transfer

For "blockwise transfer" from a computer, transfer software is required, e.g. the data transfer software from HEIDENHAIN for personal computers.

For this operating mode, you must set the following machine parameters:

MP 5030 = 1 (blockwise transfer is selected).

MP 5020 = e.g. 168 (data format).

The following machine parameters determine the control character (for description see "External Data Transfer, Machine parameters") and are valid for the data transfer software from HEIDENHAIN. If different transfer software is used, the machine parameters must be adapted correspondingly.

MP 5010 = 515

MP 5010.1 = 17736

MP 5010.2 = 16712

MP 5010.3 = 279

MP 5010.4 = 5382

MP 5010.5 = 4

When using the transfer software from HEIDENHAIN, the data interface is normally set to "FE". Then the above machine parameters need not be entered.

Adapting to non-HEIDENHAIN devices

Compare the interface descriptions of both devices.
Then proceed as follows:

- Determine the common settings (data format, baud rate).
The peripheral device is usually set via internal switches.
- Determine the pin layout for the data transfer cable, and wire the cable.
- Plug in the data transfer cable.
- Plug in the power cord of the peripheral device.
- Switch on power.
- Start the transfer software from the computer, if required.
- Select the transfer menu on the TNC with the "EXT" key and start the desired transfer.

External Data Transfer

Machine parameters



The following settings are only effective when operating the data interface in the "EXT" operating mode. To select the machine parameters, see index A "General Information, MOD Functions, User parameters".

MP 5010 Control characters for blockwise transfer

MP	Bit	Function	Input values ¹⁾
5010.0	0 ... 7 8 ... 15	ETX or any ASCII character. Character for end of program. STX or any ASCII character. Character for start of program.	ETX and STX: 515
5010.1	0 ... 7 8 ... 15	H or any ASCII character. It is sent in the command block for data input prior to the program number. E or any ASCII character. It is sent in the command block for data input after the program number.	H and E: 17736
5010.2	0 ... 7 8 ... 15	H or any ASCII character. It is sent in the command block for data output prior to the program number. A or any ASCII character. It is sent in the command block for data output after the program number.	H and A: 16712
5010.3	0 ... 7 8 ... 15	ETB or substitute character (decimal code 1–47) is sent at the end of the command block . SOH or substitute character (decimal code 1–47) is sent at the beginning of the command block .	ETB and SOH: 279
5010.4	0 ... 7 8 ... 15	ACK or substitute character (decimal code 1–47): positive acknowledgement. It is sent when the data block is correctly received. NAK or substitute character (decimal code 1–47): negative acknowledgement. It is sent when the data block is incorrectly transferred.	ACK and NAK: 5382
5010.5	0 ... 7	EOT or substitute character (decimal code 1–47) is sent at the end of the data transfer .	EOT: 4

¹⁾ The input values apply for the data transfer software from HEIDENHAIN

MP 5010.0

This defines one character from the ASCII character code for the end of program and one for the start of program for external programming. ASCII characters 1–47 are accepted.
"End of program" is sent at "standard data interface" and "blockwise transfer".
"Start of program" is only sent at "blockwise transfer".

Determining bit significance MP 5010.0

Example: End of program: ETX BINARY code 00000011
Start of program: STX BINARY code 00000010

Bits 0 – 7	7	6	5	4	3	2	1	0
Significance of bit	128	64	32	16	8	4	2	1
Enter 0 or 1 accordingly	0	0	0	0	0	0	1	1
Bits 8 – 15	15	14	13	12	11	10	9	8
Significance of bit	32768	16384	8192	4096	2048	1024	512	256
Enter 0 or 1 accordingly	0	0	0	0	0	0	1	0

Determine input value:

$$\begin{array}{r}
 1 \\
 2 \\
 + 512 \\
 \hline
 515
 \end{array}$$

The input value for MP 5010.0
is thus 515.

External Data Transfer

Machine parameters



The data format and the type of transfer stop are determined by MP 5020. Bit 1 is only set for "blockwise transfer". 0 is entered for standard data interface.

MP 5020 Data format

Function	Bit	Input	Input values										
7 or 8 data bits	0	+ 0 → 7 data bits (ASCII code with 8 th bit = parity) + 1 → 8 data bits (ASCII code with 8 th bit = 0 and 9 th bit = parity)	1										
Block Check Character (BCC)	1	+ 0 → any BCC character + 2 → BCC character no control character	–										
Transfer stop due to RTS	2	+ 0 → inactive + 4 → active	–										
Transfer stop due to DC3	3	+ 0 → inactive + 8 → active	8										
Character parity even or odd	4	+ 0 → even + 16 → odd	32										
Character parity required	5	+ 0 → not required + 32 → required											
Number of stop bits	<table><tr><td>7</td><td>6</td></tr><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	7	6	0	0	0	1	1	0	1	1	1 1/2 Stop bits 2 Stop bits bit 6: + 64 1 Stop bit bit 7: + 128 1 Stop bit	128
7	6												
0	0												
0	1												
1	0												
1	1												

value to be entered for MP 5020

169

Notes on
bit 1

Input value does not contain the significance 2:

The BCC can accept an arbitrary character (also control character) in "blockwise transfer".

Input value contains the significance 2:

If the computation of the BCC during "blockwise transfer" results in a number less than 20 HEX¹⁾ (control character), then a "space" character (20 HEX) is additionally sent prior to ETB. In this case, the BCC is always greater than 20 HEX and therefore not a control character.

¹⁾ HEX = Hexadecimal

Example
of value
determination

Standard data format:

7 data bits (ASCII code with 7 bits, even parity)

Transfer stop due to DC3, 1 stop bit

Bits 0 – 7	7	6	5	4	3	2	1	0
Significance of bit	128	64	32	16	8	4	2	1
Enter 0 or 1 accordingly	1	0	1	0	1	0	0	0

After adding the significances, you obtain the input value for machine parameter 5020.
In our example: 168.

MP 5030 Operating mode of the interface

Operating mode data interface RS-232-C

This parameter determines the function of the data interface.

0 ≙ "standard data interface" (normally for printer, reader, punch)

1 ≙ "blockwise transfer" (normally for computer link)

External Data Transfer

Machine parameters



The following settings are only effective when operating the data interface in the "EXT" operating mode. To select the machine parameters, see index A "General Information, MOD Functions, User parameters".

MP 5010 Control characters for blockwise transfer

MP	Bit	Function	Input values ¹⁾
5010.0	0 ... 7 8 ... 15	ETX or any ASCII character. Character for end of program. STX or any ASCII character. Character for start of program.	ETX and STX: 515
5010.1	0 ... 7 8 ... 15	H or any ASCII character. It is sent in the command block for data input prior to the program number. E or any ASCII character. It is sent in the command block for data input after the program number.	H and E: 17736
5010.2	0 ... 7 8 ... 15	H or any ASCII character. It is sent in the command block for data output prior to the program number. A or any ASCII character. It is sent in the command block for data output after the program number.	H and A: 16712
5010.3	0 ... 7 8 ... 15	ETB or substitute character (decimal code 1–47) is sent at the end of the command block . SOH or substitute character (decimal code 1–47) is sent at the beginning of the command block .	ETB and SOH: 279
5010.4	0 ... 7 8 ... 15	ACK or substitute character (decimal code 1–47): positive acknowledgement. It is sent when the data block is correctly received. NAK or substitute character (decimal code 1–47): negative acknowledgement. It is sent when the data block is incorrectly transferred.	ACK and NAK: 5382
5010.5	0 ... 7	EOT or substitute character (decimal code 1–47) is sent at the end of the data transfer .	EOT: 4

¹⁾ The input values apply for the data transfer software from HEIDENHAIN

MP 5010.0

This defines one character from the ASCII character code for the end of program and one for the start of program for external programming. ASCII characters 1–47 are accepted.
"End of program" is sent at "standard data interface" and "blockwise transfer".
"Start of program" is only sent at "blockwise transfer".

Determining bit significance MP 5010.0

Example: End of program: ETX BINARY code 00000011
 Start of program: STX BINARY code 00000010

Bits 0 – 7	7	6	5	4	3	2	1	0
Significance of bit	128	64	32	16	8	4	2	1
Enter 0 or 1 accordingly	0	0	0	0	0	0	1	1
Bits 8 – 15	15	14	13	12	11	10	9	8
Significance of bit	32768	16384	8192	4096	2048	1024	512	256
Enter 0 or 1 accordingly	0	0	0	0	0	0	1	0

Determine input value:

$$\begin{array}{r}
 1 \\
 2 \\
 + 512 \\
 \hline
 515
 \end{array}$$

The input value for MP 5010.0 is thus 515.

External Data Transfer

Machine parameters



The data format and the type of transfer stop are determined by MP 5020. Bit 1 is only set for "blockwise transfer". 0 is entered for standard data interface.

MP 5020 Data format

Function	Bit	Input	Input values
7 or 8 data bits	0	+ 0 → 7 data bits (ASCII code with 8 th bit = parity) + 1 → 8 data bits (ASCII code with 8 th bit = 0 and 9 th bit = parity)	1
Block Check Character (BCC)	1	+ 0 → any BCC character + 2 → BCC character no control character	–
Transfer stop due to RTS	2	+ 0 → inactive + 4 → active	–
Transfer stop due to DC3	3	+ 0 → inactive + 8 → active	8
Character parity even or odd Character parity required	4 5	+ 0 → even + 16 → odd + 0 → not required + 32 → required	32
Number of stop bits	7 6 0 0 0 1 1 0 1 1	1 1/2 Stop bits 2 Stop bits bit 6: + 64 1 Stop bit bit 7: + 128 1 Stop bit	128

value to be entered for MP 5020

169

Notes on
bit 1

Input value does not contain the significance 2:

The BCC can accept an arbitrary character (also control character) in "blockwise transfer".

Input value contains the significance 2:

If the computation of the BCC during "blockwise transfer" results in a number less than 20 HEX¹⁾ (control character), then a "space" character (20 HEX) is additionally sent prior to ETB. In this case, the BCC is always greater than 20 HEX and therefore not a control character.

¹⁾ HEX = Hexadecimal

Example
of value
determination

Standard data format:

7 data bits (ASCII code with 7 bits, even parity)
Transfer stop due to DC3, 1 stop bit

Bits 0 – 7	7	6	5	4	3	2	1	0
Significance of bit	128	64	32	16	8	4	2	1
Enter 0 or 1 accordingly	1	0	1	0	1	0	0	0

After adding the significances, you obtain the input value for machine parameter 5020.
In our example: 168.

MP 5030 Operating mode of the interface

Operating mode data interface RS-232-C

This parameter determines the function of the data interface.

0 ≙ "standard data interface" (normally for printer, reader, punch)

1 ≙ "blockwise transfer" (normally for computer link)

Address Letters in ISO

Address code	Function
%	Program start or call
A B C	(rotation about X-axis) (rotation about Y-axis) (rotation about Z-axis)
D	Parameter definition (Program parameter Q)
F F F	Feed rate Dwell with G04 Scaling factor with G72
G	Preparatory function (G code)
H H	Polar coordinate angle in incremental/absolute dimensions Angle of rotation with G73
I J K	X-coordinate of circle center/pole Y-coordinate of circle center/pole Z-coordinate of circle center/pole
L L L	Set label number with G98 Jump to label number Tool length with G99
M	Miscellaneous functions
N	Block number
P P	Cycle parameter in cycles Parameter in parameter definitions
Q	Program parameter "Q"
R R R R R	Polar coordinate radius Circle radius with G02/G03/G05 Rounding-off radius with G25/G26/G27 Chamfer length with G24 Tool radius with G99
S	Spindle speed
T T	Tool definition with G99 Tool call
U V W	Linear movement parallel to X-axis Linear movement parallel to Y-axis Linear movement parallel to Z-axis
X Y Z	X-axis Y-axis Z-axis
*	End of block

Parameter Definitions in ISO

D	Function	Reference Page
00	Assign	P 106
01 02 03 04	Addition Subtraction Multiplication Division	P 106
05	Square root	P 106
06 07	Sine Cosine	P 107
08	Root-sum of squares ($c = \sqrt{a^2 + b^2}$)	P 107
09 10 11 12	If equal, jump If unequal, jump If greater, jump If less, jump	P 109
13	Angle of $c \cdot \sin \alpha$ and $c \cdot \cos \alpha$)	P 108

G Codes

Group	G	Function	Non-modal	Reference Page
Path types	00	Linear interpolation, Cartesian, rapid traverse		P 25
	01	Linear interpolation, Cartesian		P 26
	02	Circular interpolation, Cartesian, clockwise		P 33
	03	Circular interpolation, Cartesian, counterclockwise		
	05	Circular interpolation, Cartesian, no direction specified		
	06	Circular interpolation, Cartesian, tangential transition from previous contour		P 39
	07	Paraxial positioning block	•	M 18
	10	Linear interpolation, polar, rapid traverse		P 43
	11	Linear interpolation, polar		
	12	Circular interpolation, polar, clockwise		P 44
	13	Circular interpolation, polar, counterclockwise		
	15	Circular interpolation, polar, no direction specified		
	16	Circular interpolation, polar, tangential transition from previous contour		P 45
Cycles	04	Dwell	•	P 102
	28	Mirror image		P 96
	36	Oriented spindle stop		P 104
	37	Pocket contour definition		P 78
	39	Designates program, call via G79	•	P 103
	54	Datum shift		P 94
	56	Pre-drilling (used with G37)		P 89
	57	Roughing out (used with G37)		P 78
	58	Contour milling clockwise (used with G37)		P 90
	59	Contour milling counterclockwise (used with G37)		
	72	Scaling factor		P 100
	73	Coordinate system rotation		P 98
	74	Slot milling		P 71
	75	Rectangular pocket milling clockwise		P 73
	76	Rectangular pocket milling counterclockwise		
	77	Circular pocket milling clockwise		P 75
	78	Circular pocket milling counter-clockwise		
	83	Peck drilling		P 67
	84	Tapping		P 70
	79	Cycle call	•	P 65
Selection of working plane	17	Plane selection XY, tool axis Z		P 20
	18	Plane selection ZX, tool axis Y		
	19	Plane selection YZ, tool axis X		
	20	Tool axis = 4 th axis		
Chamfer, corner rounding, approach and departure	24	Chamfer with length R	•	P 27
	25	Corner rounding with R	•	P 37
	26	Tangential contour approach with R	•	P 50
	27	Tangential contour departure with R	•	
	29	Designate last nominal value as pole	•	P 42
Blank form definition	30	Blank workpiece definition for graphics: min point		P 8
	31	Blank workpiece definition for graphics: max. point		
	38	STOP program run	•	P 20
Tool path compensation	40	No tool compensation (R0)		P 15
	41	Tool path compensation, left of contour (RL)		
	42	Tool path compensation, right of contour (RR)		
	43	Paraxial compensation extension (R+)		P 17
	44	Paraxial compensation, reduction (R-)		
	50	Program protection (at start of program)		P 7
	51	Next tool number (when using central tool memory)	•	
	55	Touch probe function	•	P 119
Unit of measure	70	Dimensions specified in inches (at start of program)		P 6
	71	Dimensions specified in millimeters (at start of program)		
Dimensions	90	Absolute dimensions		A 17
	91	Incremental dimensions		
	98	Set label number	•	P 56
	99	Tool definition	•	P 10

M Functions

Miscellaneous functions with predetermined function

M	Function	Effective		Reference Page
		Begin of block	End of block	
00	Stop program run/stop spindle/coolant off		•	
02	Stop program run/stop spindle/coolant off/ or clear the status display/return jump to block 1		•	P 21
03	Spindle on clockwise	•		
04	Spindle on counterclockwise	•		
05	Spindle stop		•	
06	Tool change/or stop program run/stop spindle		•	
08	Coolant on	•		
09	Coolant off		•	
13	Spindle on clockwise/coolant on	•		
14	Spindle on counterclockwise/coolant on	•		
30	like M02		•	P 21
89	Vacant miscellaneous function or	•		
89	Cycle call, modal (depends on machine parameters)		•	P 65
90	Constant path speed on inside corners and uncompensated corners	•		P 51
91	In the positioning block: Coordinates refer to the machine datum	•		P 54
92	In the positioning block: Coordinates refer to an additional machine datum defined by the machine manufacturer, such as tool change position	•		P 54
93	Reserved	•		
94	Reduce the display of the rotary axis to a value under 360°	•		
95	Reserved		•	
96	Reserved		•	
97	Path offset on outside corners: Intersection instead of tangential circle		•	P 52
98	Blockwise end of path offset		•	P 53
99	Cycle call effective blockwise		•	P 65

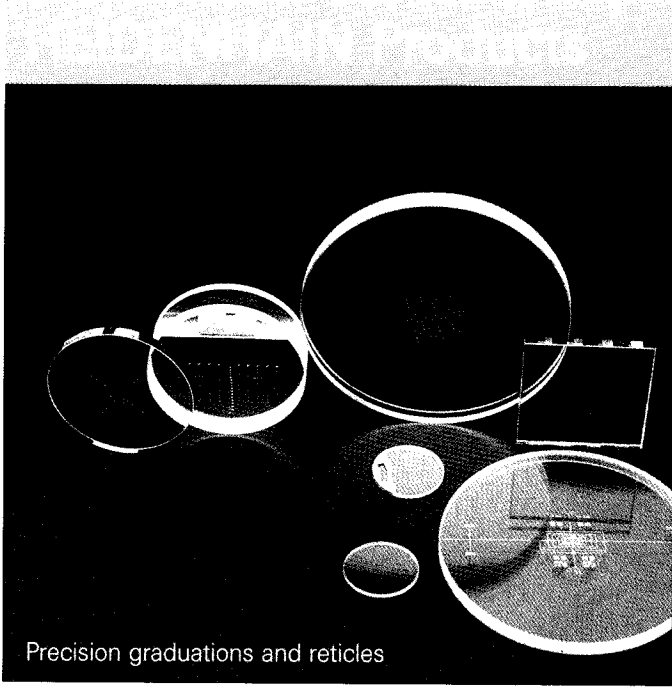
M Functions

Vacant miscellaneous functions

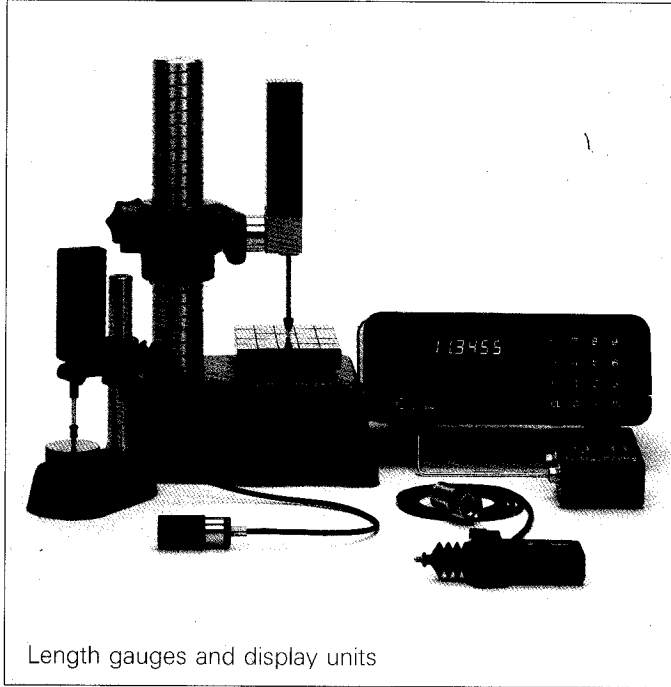
M	Function	Effective at	
		Begin of block	End of block
01			•
07		•	
10			•
11		•	
12			•
15		•	
16		•	
17		•	
18		•	
19			•
20		•	
21		•	
22		•	
23		•	
24		•	
25		•	
26		•	
27		•	
28		•	
29		•	
31		•	
32			•
33			•
34			•
35			•
36		•	
37		•	
38		•	
39		•	
40		•	
41		•	
42		•	
43		•	
44		•	
45		•	
46		•	
47		•	
48		•	
49		•	
50		•	
51		•	

M	Function	Effective at	
		Begin of block	End of block
52			•
53			•
54			•
55		•	
56		•	
57		•	
58		•	
59		•	
60			•
61		•	
62		•	
63			•
64			•
65			•
66			•
67			•
68			•
69			•
70			•
71		•	
72		•	
73		•	
74		•	
75		•	
76		•	
77		•	
78		•	
79		•	
80		•	
81		•	
82		•	
83		•	
84		•	
85		•	
86		•	
87		•	
88		•	

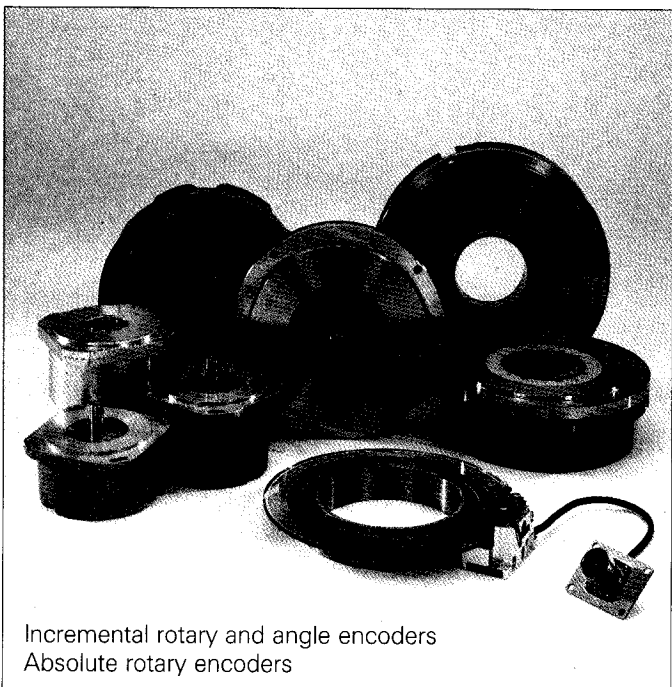
These miscellaneous functions are assigned by the machine tool builder and are described in the operating instructions for your machine tool.



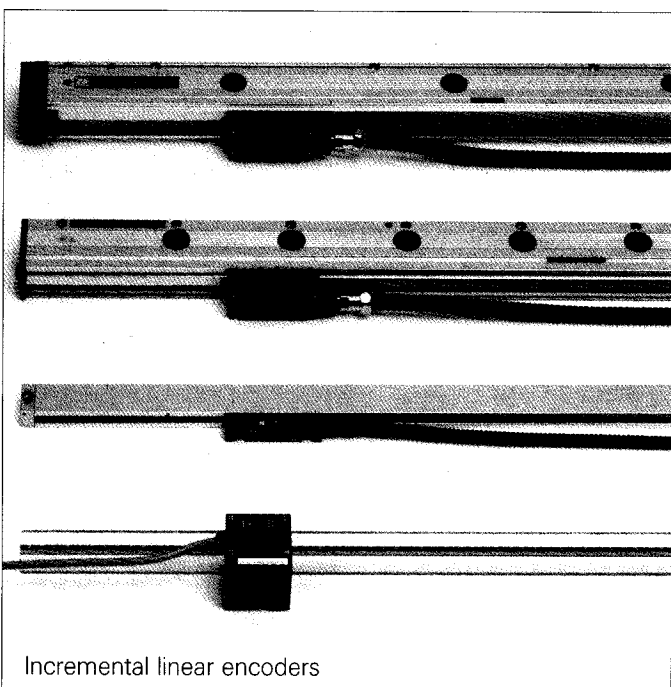
Precision graduations and reticles



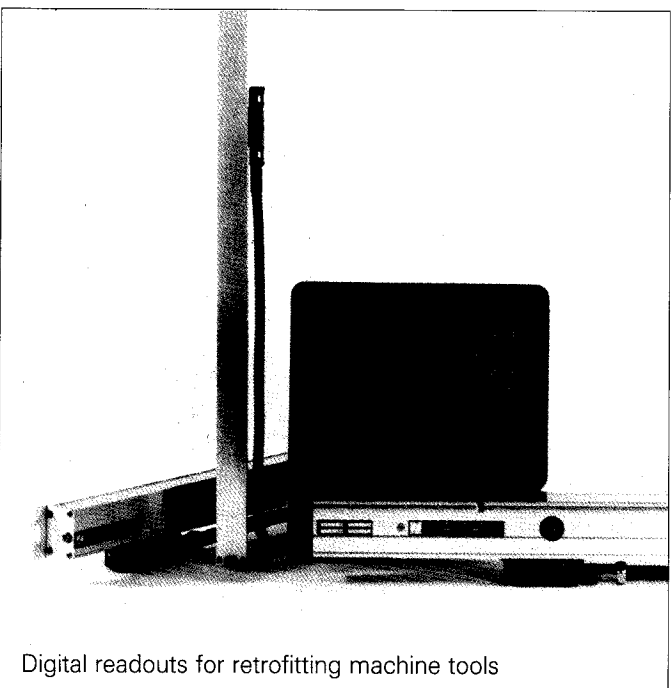
Length gauges and display units



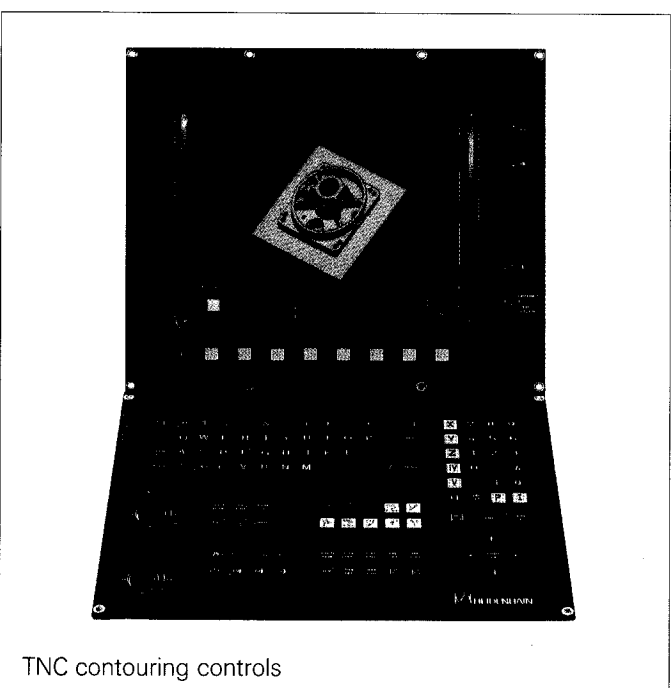
Incremental rotary and angle encoders
Absolute rotary encoders



Incremental linear encoders



Digital readouts for retrofitting machine tools



TNC contouring controls



HEIDENHAIN

Contouring Control

Working plane:		
Tool axis (90°)	Working plane	Reference axis (0°)
Z (G17)	XY	X
X (G18)	YZ	Y
Y (G19)	ZX	Z

Tool radius compensation:

Contour	Order of programmed contour elements	Radius compensation
inner (pocket)	clockwise (CW) counterclockwise (CCW)	G42 (RR) G41 (RL)
outer (island)	clockwise (CW) counterclockwise (CCW)	G41 (RL) G42 (RR)

Cycles:

G	Cycle	effective after call-up	effective immediately
83	Pecking	•	
84	Tapping	•	
74	Slot milling	•	
75/76	Pocket milling	•	
77/78	Circular pocket	•	
39	Program call	•	
37	Define contour		•
56	Pilot drilling	•	
57	Rough-out	•	
58/59	Contour milling	•	
54	Datum shift		•
28	Mirror image		•
73	Rotation		•
72	Scaling factor		•
04	Dwell time		•

Contour Cycles:

Program structure when machining with several tools

List of contour subprograms	G37 P01 ...
Drill define/call	
Contour cycle: pilot drilling	G56 P01 ...
Pre-position, cycle call	
Roughing cutter define/call	
Contour cycle: rough-out	G57 P01 ...
Pre-position, cycle call	
Finishing cutter define/call	
Contour cycle: contour milling	G58 P01 ...
Pre-position, cycle call	
End of main program, return	M02
Contour subprograms	G98
	G98 L0

Coordinate Transformations:

Coordinate transformation	Activate	Cancel
Datum shift	G54 X+20 Y+30 Z+10	G54 X+0 Y+0 Z+0
Mirror image	G28 X	G28
Rotation	G73 H+45	G73 H+0
Scaling factor	G72 F0,8	G72 F1

Program Section Repeat:

Label number identifies program section to be repeated
Repeat five times = execute six times

G98 L2
G00 G91 X+10 M99
L2,5

Subprogram:

Subprogram call
L4,0
M02 indicates end of main program and return to beginning of program
Beginning of subprogram
G98 L4
End of subprogram (Further subprograms)
G98 L0

Program call:

Call another program with



Helical interpolation:

Incremental value $H = 360 \cdot \frac{Z}{P}$
Z = incremental advance
P = thread pitch

Approach starting position
Define pole
Helical interpolation
H+50 J+30
G13 G41 G91 H-2520 Z+12

Right-hand thread		Left-hand thread		Direction of rotation	Working direction of tool axis
inner	outer	inner	outer		
G41 (RL)	G42 (RR)	G42 (RR)	G41 (RL)	clockwise (CW)	negative
G42 (RR)	G41 (RL)	G41 (RL)	G42 (RR)	counterclockwise (CCW)	positive

Program Structure

Select program number

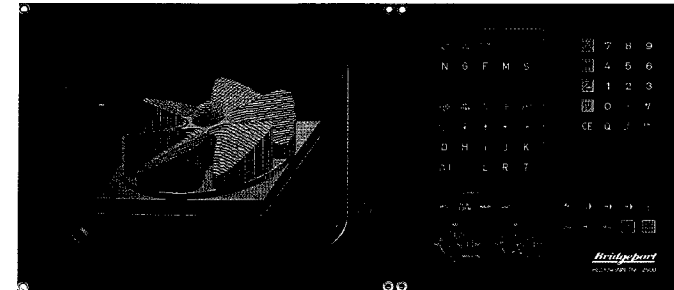


Program 234 in mm
Blank form definition
% 234 G71
G30 G17 X+0 Y+0 Z-40
G31 G90 X+100 Y+100 Z+0
Tool definition
G99 T1 L+0 R+5
Tool call
T0 G17
Tool change position
G00 **G40** G90 Z+100 **M06**
Tool call
T1 G17 S1000
Starting position, next to the workpiece
X-20 Y-20 M03
Working depth
Z-20
1st contour point, with compensation (RL)
G01 **G41** X+0 Y+0 F200
Tangential approach
G26 R15
Straight line
Y+100
Chamfer
G24 R20
Straight line
X+100
Rounding
G25 R20
Straight line
Y+25
Circle center
H+100 J+0
Circle, incremental
G03 G91 X-25 Y-25
Last contour point, absolute
G01 G90 X+0 Y+0
Tangential departure
G27 R15
End position, next to the workpiece
G00 **G40** X-20 Y-20
Retract, return to beginning of program
Z+100 **M02**



HEIDENHAIN

TNC 2500 B Contouring Control



Operating Modes

Machine control:



Manual

The axes can be moved via the external axis direction buttons. The position displays can be set to desired values.



Electronic Handwheel

The axes can be driven either by using the electronic handwheel or by entering a jog increment via the external axis direction keys.



Positioning via MDI

The axes are moved **to** or **by** a manually-entered dimension with the chosen radius compensation, feed rate and M function. The block is not stored!



Program Run, Full Sequence

After start of the program via the external START key, the program will automatically be executed up to the end of the program or STOP.



Program Run, Single Block

Any single block can be started separately via the external START key.

Programming:



Programming & Editing

Part programs can be entered, checked and altered. They can also be read-in and read-out via RS-232-C/V.24 interface.



Test

Part programs are tested for logical errors such as machine traverse limit violations, double programming of axes etc.

Test Graphics:



GRAPHICS

Part programs are graphically simulated in plan view, projection in three plains and 3D view. This test run is conducted in the "full block" and "single block" operating modes and is started with the "START" key on the control keyboard.

Adresses

Address	Function	Address	Function
%	Program beginning	P	Cycle parameter for fixed cycles
%	Programm call with G39	P	Value or Q-parameter in Q-parameter definition
A	Rotary axis about X-axis	Q	Q-parameter
B	Rotary axis about Y-axis	R	Radius for polar co-ordinates
C	Rotary axis about Z-axis	R	Circle radius with G02/G03/G05
D	Q-Parameter definitions	R	Round-off radius w. G25/G26/G27
F	Feed rate	R	Tool radius with G99
F	Dwell time with G04	S	Spindle speed
F	Scaling factor with G72	T	Tool definition with G99
G	G functions	T	Tool call
H	Angle for polar co-ordinates	T	Next tool with G51
H	Rotational angle with G73	U	Additional linear axis parallel to X-axis
I	X-Coordinate of circle centre/pole	V	Additional linear axis parallel to Y-axis
J	Y-Coordinate of circle centre/pole	W	Additional linear axis parallel to Z-axis
K	Z-Coordinate of circle centre/pole	X	X-Axis
L	Set label number with G98	Y	Y-Axis
L	Jump to label number	Z	Z-Axis
L	Tool length with G99	*	End of block
M	Miscellaneous (M) functions		
N	Block number		

Q-Parameter Definitions

D	Function
00	Allocation
01	Addition
02	Subtraction
03	Multiplication
04	Division
05	Root
06	Sine
07	Cosine
08	Root sum of squares ($c = \sqrt{a^2 + b^2}$)
09	If equal, jump to label number
10	If not equal, jump to label number
11	If greater, jump to label number
12	If less, jump to label number
13	Angle (from $c \cdot \sin \alpha$ and $c \cdot \cos \alpha$)

G-Codes

G	Function	effective blockwise
00	Linear interpolation, Cartesian, in rapid traverse	
01	Linear interpolation, Cartesian	
02	Circular interpolation, Cartesian, clockwise (CW)	
03	Circular interpolation, Cartesian, counterclockwise (CCW)	
05	Circular interpolation, Cartesian, without direction data	
06	Circular interpolation, Cartesian, tangential	
07	Single axis positioning block	•
10	Linear interpolation, polar, in rapid traverse	
11	Linear interpolation, polar	
12	Circular interpolation, polar CW	
13	Circular interpolation, polar CCW	
15	Circular interpolation, polar, without direction data	
16	Circular interpolation, polar, tangential contour approach	
04	Dwell	•
28	Mirror image	
37	Definition of contour cycle	
39	Designates program for call-up via G79	•
54	Datum shift	
56	Pilot drill (with G37)	
57	Rough-out (with G37)	
58	Contour mill CW (with G37)	
59	Contour mill CCW (with G37)	
72	Scaling factor	
73	Co-ordinate system (plane) rotation	
74	Slot milling	
75	Rectangular pocket milling CW	•
76	Rectangular pocket milling CCW	•
77	Circular pocket milling CW	•
78	Circular pocket milling CCW	•
83	Pecking	
84	Tapping	
79	Cycle call	•
17	XY-plane designation, Tool axis Z	
18	ZX-plane designation, Tool axis Y	
19	YZ-plane designation, Tool axis X	
20	FOURTH tool axis	
24	Chamfer with R	•
25	Rounding of corners with R	•
26	Tangential contour approach (run-on) with R	•
27	Tangential contour departure (run-off) with R	•
29	Transfer of last nominal position value as pole	
30	Blank form definition for graphics (G17/G18/G19) – min. point	
31	Blank form definition for graphics (G90/G91) – max. point	
38	Program run STOP	•
40	No tool compensation (R0)	
41	Tool radius compensation to contour, offset left (RL)	
42	Tool radius compensation to contour, offset right (RR)	
43	Tool length compensation positive (R+)	
44	Tool length compensation negative (R–)	
50	Erase/Edit protection (at program beginning)	
51	subsequent tool number (with central tool memory)	•
55	Touch probe function	•
70	Dimensioning in inches (at program start)	
71	Dimensioning in millimetres (at program start)	
90	Absolute dimensioning	
91	Incremental dimensioning	
98	Assign label number	•
99	Tool definition	•

M Functions

M	Function	Effective at block begin.	Effective at block end
00	Program run stop/spindle stop/coolant off		•
02	Program run stop/spindle stop/coolant off clear status display Return jump to first program block		•
03	Spindle on, clockwise (CW)	•	
04	Spindle on, counter-clockwise (CCW)	•	
05	Spindle stop		•
06	Tool change/program run stop/spindle stop		•
08	Coolant on	•	
09	Coolant off		•
13	Spindle on, clockwise/coolant on	•	
14	Spindle on, counter-clockwise/coolant on	•	
30	As per M02		•
89	Vacant miscellaneous function or	•	
89	Cycle call, modally effective (depending on the machine parameters entered)		•
90	Constant path feed rate on internal corners and uncompensated corners	•	
91	Within a positioning block: Workpiece zero datum is replaced by reference point	•	
92	Within a positioning block: The set workpiece zero datum is replaced by a position which is defined by the machine tool builder using a machine parameter, (e. g. tool change position)	•	
93	Reserved	•	
94	Rotary table axis display reduction to a value below 360°		
95	Reserved		•
96	Reserved		•
97	Path offset on external corners: point of intersection instead of transitional circle		•
98	End of path offset blockwise		•
99	Cycle call blockwise		•